UNIVERSITÀ DEGLI STUDI DI MILANO

CORSO DI DOTTORATO
Informatica – Ciclo XXXVII

DIPARTIMENTO DI AFFERENZA
Dipartimento di Informatica "Giovanni Degli Antoni"

TESI DI DOTTORATO DI RICERCA
Modeling Semantic Change through Large Language Models

SETTORE SCIENTIFICO DISCIPLINARE
Informatica – INF/01

DOTTORANDO
Francesco Periti

TUTOR
Stefano Montanelli

COORDINATORE DEL DOTTORATO
Roberto Sassi

ANNO ACCADEMICO
A.A. *2023-2024*

Modeling Semantic Change through Large Language Models

Francesco Periti

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

University of Milan
2024

Supervisor:
Prof. Stefano Montanelli
University of Milan, Computer Science

Ph.D. coordinator:
Prof. Roberto Sassi
University of Milan, Computer Science

Reading Committee:
Prof. Danushka Bollegala
University of Liverpool, Computer Science

Prof. Pierpaolo Basile
University of Bari Aldo Moro, Computer Science

Prof. Simon Hengchen
University of Geneva, Translation Technology

Ph.D. program in Computer Science
Departmen of Computer Science Cycle XXXVII
Section INF/01

University of Milan

Abstract

Modeling Semantic Change through Large Language Models

Francesco Periti

In recent years, Natural Language Processing has gained increasing attention due to the unprecedented capabilities of *large language models* in facilitating linguistic analyses of human language. Among these analyses, the digitization of text corpora has recently prompted the use of language models to support and automate the study of language from a diachronic perspective. Language is viewed as a dynamic entity over time where words can undergo *semantic change*, i.e., changes in their meaning and interpretation.

This thesis is about the modeling of semantic change over text corpora using large language models. Specifically, it primarily addresses a type of semantic change known in Linguistics as *lexical semantic change*, where individual words change in meaning over time. In this regard, we explore the following research questions.

- Large language models represent state-of-the-art solutions in nearly all Natural Language Processing downstream tasks. Thus, *how can lexical semantic change be modeled using large language models?*

- Lexical semantic change is typically modeled across two time periods. Thus, *how can the existing modeling be expanded to handle multiple time periods?*

- The current modeling of semantic change focuses on word-level granularity (i.e., lexical semantic change). Thus, *how can the existing modeling be extended to address text-level semantic change?* Specifically the phenomenon known in Linguistics as *historical resonance*.

First, we comprehensively review the state-of-the-art research on lexical semantic change and propose a framework for classifying different approaches that use large language models. We outline the effectiveness and limitations of these approaches and identify several open challenges in the current modeling. Throughout this thesis, we extend the existing computational task of detecting lexical semantic change by integrating it with other relevant, related tasks, such as modeling semantic judgments of words in-context (also known as Word-in-Context) and modeling the meaning of words (also known as Word Sense Induction). To this end, we explore different semantic representations of word meaning, including word embeddings, lexical replacements, and sense definitions. We evaluate state-of-the-art approaches and propose multiple solutions, each with distinct benefits and limitations. Considering word embeddings, we find that monolingual pre-trained

BERT models outperform multilingual pre-trained models such as mBERT and XLM-R for modeling semantic change. Additionally, we discovered that the standard practice of using word embeddings generated by the last layer of these models is typically not the most effective option for modeling semantic change. Instead, we found that other layers consistently achieve higher performance. Furthermore, we find that approaches that quantify semantic change based on features such as polysemy and dominant word meaning prove to be more powerful than those attempting to model each meaning of a word individually before modeling semantic change. Finally, given that word embeddings often pose interpretability issues, we also demonstrate that lexical replacements and sense definitions automatically generated by Llama and Flan-T5 models are interpretable and promising solutions for modeling lexical semantic change.

Considering the second research question, we extend the current modeling of lexical semantic change from two time periods to multiple time periods. This extension allows us to capture the evolution of each individual sense of a word over time. In this regard, we outline different strategies for extending the current modeling and present a novel, scalable, and evolutionary clustering algorithm for modeling word meaning over time. Through rigorous experimentation, we demonstrate the effectiveness of this algorithm in general clustering settings. We then integrate it into a novel approach for modeling lexical semantic change and evaluate its use against established benchmarks and across different languages. Finally, we illustrate its application by analyzing target words across two Italian datasets containing Italian parliamentary speeches and Vatican publications.

In the last part of this thesis, we extend the current modeling of semantic change from lexical semantic change to historical resonance. Thus far, historical resonance has been modeled by merely considering the detection of text reuse excerpts (e.g., literary quotations). However, we observe that these approaches do not focus on *recontextualization*, i.e., how the new context(s) of a reused text differs from its original context(s). We thus define *historical resonance* as text-reuse *re-contextualization* and introduce a novel evaluation framework to evaluate computational methods in capturing the *recontextualization* of text-reuse. This framework relies on the notion of topic relatedness for evaluating the diachronic change of context in which text is reused. We conduct a human-annotation campaign to create an evaluation benchmark with gold labels of topic relatedness. Then, we comprehensively evaluate a set of SBERT models to assess their suitability for modeling historical resonance through topic relatedness of text reuse. Our experiments show that these models exhibit greater sensitivity to textual similarity rather than topic relatedness, and that fine-tuning these models can mitigate such a kind of sensitivity.

Overall, this thesis contributes to the growing field of Natural Language Processing and Computational Linguistics, advancing the state-of-the-art in computational modeling of semantic change. By addressing key research questions and proposing innovative methodologies, we provide valuable insights and tools for modeling the dynamic nature of word and text semantics, and its evolution over time.

Acknowledgements

To those of you reading these acknowledgments, you must know that this PhD journey has been *crazy*. There is no semantic change in the word *crazy*; I used it literally on purpose as I have almost gone insane. It was a long journey filled with frustration, discussions, and arguments. At times, I lost motivation and hope in my research, and I felt completely alone. But now it's finished, and I am extremely excited to see my research with new eyes. This is probably the same for every PhD story. However, I often felt like an extraordinary case and would have probably given up if I had not met special people and collaborators.

As this thesis closes an important chapter of my life, I want to express my most sincere gratitude to those without whom it would not have been possible. This includes everyone who contributed to the realization of this work, but especially those who were my supporters, listeners, advisors, and family during both difficult and easier times throughout this PhD.

Thus, many thanks to my supervisor, Stefano Montanelli. Thank you for your patience. I hope we both learned something from our fights and discussions: they have been valuable, both when you succeeded in changing my perspective and when you reinforced my own vision even further. Thank you for teaching me how to write scientific papers. I will long think of you when I write in English, especially about your favorite patterns and words that you would probably like me to use even in these acknowledgments. Thanks for allowing me to participate in multiple internships: they helped me grow both personally and professionally as a researcher. Finally, special thanks for letting me spend so much time abroad, both officially and unofficially, to work close to my wife: *this was fundamental for finishing my PhD*.

My second acknowledgment, but by no means less important, goes to Nina Tahmasebi. I truly believe I owe you this PhD. It has not been easy, but you definitely made it as painless as possible. When we met, my English was so bad, and I was ashamed of it. Thank you for making me feel important and skillful from the start of our collaboration, for having the patience to listen to my slow and incorrect speech without highlighting my weaknesses or showing any judgment. Thanks for all the inspiring feedback and for allowing me to lead my research, steering it in the right direction every time I faced a challenge. Your consistently kind leadership offers a safe and positive work environment, and I am immensely grateful to be a part of it and to work with you. I could continue with many other appreciations and acknowledgments but let me summarize by saying this: *you are the kind of scientist I would like to become*.

I am also deeply grateful to all the colleagues and researchers of the Change is Key! program who welcomed me on board and provided invaluable help, suggestions, and comments. To all my colleagues,

especially those with whom I encountered misunderstandings and arguments: *your collaboration has taught me what it means to be a scientist*. Among them, I must extend my thanks to Haim Dubossarsky. Thank you for believing in me when I didn't even believe in myself: *your words have been so powerful*. I hope your words have the same motivational impact on all your students. Despite some misunderstandings and conflicting schedules that paused our collaboration, I enjoyed being supervised by and working with you during my time in London and remotely in Gothenburg.

Among other scholars whom I cannot list here to avoid writing a second thesis, I want to express my gratitude to Martin Ruskov for being a special colleague. Your friendship over these years has been truly inspiring. You have consistently made time for our conversations, listened to my concerns even when I was abroad, providing support and advice. I regret that the timing for scientific collaboration wasn't right, but I hope opportunities will arise in the future.

Finally, I want to thank everyone who supported me in this work, mentally and personally. I owe you all an apology for sometimes appearing different from who I actually am. My initial difficulty with speaking English prevented me from being as friendly as I would have liked to be, and the challenges of this PhD journey often made me appear more stressed, angry, or busy than usual.

My deepest thanks go to my friends and family. Moving to another country, participating in internships across Europe, and working mostly alone remotely challenged my social life. I am deeply grateful to the Italian community in Leuven, who warmly welcomed Martina and me. Sharing our lives and experiences with you was invaluable in the last few years. I also want to express my gratitude to long-time friends for standing by me when I was scared of the *big steps* I was about to take, and for the respect, consideration, and esteem you have shown me. Although I am sorry that we are not physically close, I now appreciate our friendship not as something to take for granted, as we often do, but from a new perspective that truly makes me aware of your love. I am so proud of this that I cannot fully express it to you.

Mom, Dad, sisters, *nonna* Carla, and Martina's family, thank you all for the immense and lovely support throughout these years. You play a crucial role in my life, far beyond my research project: *simply put, I love you all deeply*.

Lastly, Martina, beyond being my partner, you have been my only listener, friend, and source of comfort during many challenging days throughout this PhD. You are the only person who truly knows the challenges, both big and small, that I have encountered over these years. You never complained about me venting my frustrations, even when you faced greater challenges than mine. Marrying you has undoubtedly been the best thing about this PhD. Striving to be the best version of myself with you is the most meaningful research I have pursued and will continue to pursue, regardless of my future in academia. I wish us a stable and happy life together, with children, and all the best in this *crazy* world.

DEDICATION

To my sisters,
Margherita, Benedetta, and Maria.

I love you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> "*The semantic fabric of the text, like the fabric of the universe, can be theorized as a space-time continuum, alive with memory of probabilities, memory of alternatives, and memory of change*"
>
> Wai Chee Dimock, *A Theory of Resonance*

Computer Science and Linguistics intersect in the field of Natural Language Processing (NLP), where algorithms and computational models are employed to analyze, interpret, and generate human language. In recent years, NLP has gained increasing attention due to the unprecedented capabilities of language models in facilitating linguistic analyses. Among these analyses, the digitization of historical text corpora has recently prompted the use of language models to support and automate the study of language from a *diachronic* perspective. Language is viewed as a dynamic entity subjected to *semantic change* in meaning and interpretation *over time* and among its users (Campbell, 2020).

Semantic change has long been studied by linguists and other scholars in the humanities through time-consuming manual activities (Blank, 1997; Bloomfield, 1933). For instance, conventional methods for detecting, interpreting, and assessing semantic change primarily rely on "close reading" and require arranging hypotheses and testing procedures to build extensive catalogs of word descriptions. These analyses keep humans "in-the-loop" and have thus been narrowed in terms of the volume, genres, and time frame that can be manually considered.

Modeling semantic change through the novel advancements in NLP presents a new opportunity to expand and scale up the analysis. Such *computational modeling of semantic change* is the central focus of this PhD thesis. Given the expansive range of computational solutions, my PhD specifically targets the modeling of semantic change through the very cutting-edge solution at its starting time, i.e., Large Language Models (LLMs) based on the Transformer architecture (Vaswani et al., 2017).

Addressing research problems in a systematic manner often involves progressing step by step, moving from smaller, more manageable units to larger and more complex components. In the context of language analysis, the smallest meaningful unit is generally considered to be a *word token*. Therefore, the primary focus of this thesis lies in modeling semantic change at the word level – i.e., modeling how words change meaning over time, a linguistic phenomenon commonly referred to as "lexical semantic change" (Geeraerts, 2020; Grondelaers et al., 2010; Bloomfield, 1933). An example of this phenomenon can be observed in the Italian words `presidente` and `presidentessa` that changed from meaning *male president* and *wife of the president* to encompass broader meanings of *president of either sex* and *female president*, respectively. [1] This transformation was prompted by 20th-century movements that advocated for women's rights to exercise professional roles with full legal and economic equality, while criticizing the derivation of female professional names from their male counterparts. Modeling such lexical semantic change represents a significant challenge that involves both distinguishing all the senses of a word and tracing their evolution over time.

As such, modeling the semantic change of a small language unit like a word token has proven to be complex, intricate, and time-consuming. As a result, during this PhD, the majority of the focus has been on word level. In the last chapter, however, we have expanded from the modeling of lexical semantic change to the modeling of "historical resonance", i.e., the linguistic phenomenon of how *well-known* text (e.g., literary text, quotes, idioms) *sounds when it is read twenty years, two hundred years, or two thousand years after it was written* (Dimock, 1997). An example of this phenomenon can be observed in the quote `To be or not to be` where Hamlet originally reflected on the struggles of existence and the fear of the unknown, contemplating the existential question of life and death. Over the centuries, the phrase has become deeply embedded in various languages and cultures, often improperly referenced, quoted, and parodied in diverse literary works, contexts, and topics (Bate, 1985). While the modeling of such kind of semantic change takes up only one chapter of this thesis, this work serves as an initial, but substantial, foundation for furthering the modeling of the NLP research community.

1.1 Motivations

From a computational perspective, an initial question that may arise is: *why engage in the modeling of semantic change?* The immediate motivation behind employing computational methods for studying semantic change lies in their ability to support text-based researchers. A reliable computational approach that efficiently analyzes vast amounts of text with limited human intervention would be an extremely useful tool to assist researchers such as linguists, historians, and lexicographers. Such a tool would assist in creating and updating linguistic resources (e.g., lexicons, vocabularies, and thesauri) while also enhancing our understanding of historical and societal change reflected in language. For instance, consider the current attention to topics like "politically correct": the word `retarded` has undergone semantic change over time, originally describing a neutral medical condition, but later acquiring offensive connotations when used as a derogatory insult (Halmari, 2011; O'Neill, 2011). This also highlights the importance of understanding and modeling semantic change to guide future changes in culture and society.

[1] accademiadellacrusca.it/it/consulenza/la-presidente-dellaccademia-della-crusca-ancora-sul-femminile-professionale/250

Computational modeling of semantic change plays a crucial role in supporting lexicographers in creating and updating linguistic resources such as lexicons, vocabularies, and thesauri. Traditionally, these resources are "synchronic", offering a perspective on language at a particular point in time, due to the meticulous manual efforts involved in their creation and updating. The adoption of computational solutions facilitates the development of more comprehensive "diachronic" resources, offering a perspective on language evolution over time, space, and communities.

Moreover, modeling semantic change represents a significant challenge in NLP. Modeling lexical semantic change, for example, serves as an important testing scenario to assess the capability of state-of-the-art language models in accurately capturing meaning in text (Periti and Montanelli, 2024). While contemporary LLMs are pre-trained on expansive all-purpose corpora, often emphasizing web corpora, researchers and practitioners employ them for diverse text applications, irrespective of the alignment between the information and language in the studied text and the pre-training text. As a matter of fact, these models serve as the interpretative lens through which we analyze the studied texts. Thus, when they are applied to study historical or other out-of-domain corpora, there could be a gap of arbitrary size that negatively impacts follow-up studies. For example, a modern, "gender-inclusive" LLM trained on contemporary text might misinterpret the Italian expression `il presidente e la presidentessa` in historical documents, interpreting it as two presidents (one male and one female) rather than as *the president and his wife*.

Finally, methods for modeling semantic change prove useful for several real-world applications. For example, integrating these methods into information retrieval and question-answering systems could enhance the user experience in information search. Traditional approaches to information retrieval rely on strategies such as adding, dropping, and substituting query terms, assuming static word meanings. However, such approaches can impact the scope and meaning of original research when users' queries or corresponding answers are affected by semantic change (Engerer, 2017). Modeling semantic change has also relevance in biomedical and clinical NLP and studies (Preiss, 2024; Xiao et al., 2023; Peterson and Liu, 2021; Yan and Zhu, 2018; Kay, 1979). For example, Preiss (2024) leverages computational models of semantic change to identify drugs suitable for repurposing. Specifically, they analyze temporal changes in word contexts to uncover new therapeutic applications for existing drugs and their compounds.

1.2 Research questions

The computational modeling of *semantic change* has witnessed a rapid evolution in the scientific literature throughout the composition of this thesis. Over the last five 5 years, the advent of the first ACL workshops on Historical Language Change (Tahmasebi et al., 2024, 2023, 2022b, 2021b, 2019) and the design of new shared tasks on LSC (Zamora-Reina et al., 2022b; Kutuzov and Pivovarova, 2021c; Basile et al., 2020; Schlechtweg et al., 2020) have sparked increasing interest among researchers and practitioners in the field of NLP. Despite this notable progress, significant open questions and challenges remain. In this regard, this thesis aims to address the following research questions (RQs) (Periti, 2023):

RQ1: *How can lexical semantic change be modeled using LLMs?*

When the work on this thesis started, computational modeling of lexical semantic change was still in its early stage. Less than a year had passed since the introduction of the first evaluation framework at the SemEval-2020 challenge (Schlechtweg et al., 2020). Advances in NLP were also younger: word embeddings generated by encoder-based LLMs (e.g., BERT) were considered the most powerful tool for representing word meanings in NLP, despite concerns about the size and number of parameters in these models. A comprehensive study of these LLMs for modeling lexical semantic change was of paramount importance to extend previous surveys on static word embeddings (Tahmasebi et al., 2021a; Kutuzov et al., 2018).

Thus, we systematically review the computational modeling of lexical semantic change using encoder-based LLMs. While exploring solutions based on these models, a novel and deeper class of generative LLMs emerged (e.g., GPT-4), showing even more interesting and promising capabilities. However, the rapid advancements in the field of NLP (Torfi et al., 2021) mean that the life of a PhD student (mine is 3 years) is too short to explore deeply and extensively every new solution. To remain current with these advancements, we dedicate three chapters to investigate the use of more recent generative LLMs.

RQ2: *How can the existing modeling be expanded to handle multiple time periods?*

With the SemEval-2020 challenge, the complexity of modeling lexical semantic change was simplified to its core due to the substantial annotation efforts required to create reliable benchmarks. Specifically, given a word, the evaluation framework involved quantifying the extent to which that word changed in meaning *over two time periods*. While this simplification served as a foundational building block of the modeling, a more complete modeling requires considering each individual meaning of a word across multiple time periods of interest.

Thus, we first connect the current LSC modeling over two time periods with other established NLP problems, such as assessing the similarity between word usages (also known as “Word-in-Context”, Cassotti et al., 2023b; Martelli et al., 2021; Liu et al., 2021a; Loureiro et al., 2022; Raganato et al., 2020; Pilehvar and Camacho-Collados, 2019), and distinguishing between different word meanings (also known as “Word Sense Induction”, Aksenova et al., 2022; Manandhar et al., 2010; Agirre and Soroa, 2007). Then, we propose various theoretical approaches to advance the current LSC modeling over multiple time periods and implement a new solution based on one of these approaches.

RQ3: *How can the existing modeling be extended to model historical resonance?*

While an evaluation framework for LSC has been established since 2020, there is no well-established evaluation framework or modeling of *historical resonance* in the present scientific literature. This complex form of text-level semantic change beyond the word level has thus far been operationalized and referred to as "text reuse", i.e. *the reuse of prior text in different sources over time* (MacLaughlin et al., 2021; Smith et al., 2014; Clough et al., 2002). Although several approaches have been proposed to *detect* text-reuse instances, they are mostly confined to *lexical matching* and do not focus on *semantic change*. As a result, the modeling of text reuse is merely approached from a computational perspective, without exploring linguistic phenomena related to variations in semantics or interpretation. These gaps render the computational modeling of text reuse in NLP a significant open problem.

Thus, in this thesis, we define *historical resonance* as text-reuse *re-contextualization* – i.e., how the new context(s) of a reused text *resonates* (i.e., differs) compared to its original context(s) – and introduce a novel evaluation framework and benchmark to advance current NLP modeling of semantic change.

1.3 Thesis outline

For the sake of simplicity, Table 1.1 offers a comprehensive overview illustrating the discourse surrounding the defined research questions throughout the entire thesis. The structure of the thesis is as follows.

Chapter 1 has so far presented the perspective and motivation that underpin this thesis.

Chapter 2 provides an original review of computational modeling of lexical semantic change at the beginning of this thesis following the advent of LLMs. In this chapter, we first define the adopted terminology and formalize the modeling. Then, we introduce a novel classification framework to survey and compare the existing state-of-the-art approaches. Finally, we discuss the main challenges and issues related to the presented modeling.

Chapter 3 offers a very first evaluation of the most recent ChatGPT model available at the time, in order to elucidate its potential as off-the-shelf model for modeling lexical semantic change. In this chapter, we first evaluate ChatGPT to detect semantic change in Word-in-Context settings under various conditions. Then, we compare its performance against a pre-trained BERT model.

Chapter 4 discusses the extension of the current modeling of lexical semantic change. In this chapter, we first outline the simplification of the existing models over two time periods and then propose approaches to advance the modeling by considering the semantics of individual words at all the relevant time points.

Chapter 5 follows the previous discussion and proposes a novel incremental clustering algorithm to distinguish the different meanings of a word by considering the temporal nature of language. In this chapter, we first present our novel algorithm, called A-Posteriori affinity Propagation (APP). Then, we evaluate its performance against conventional algorithms on standard clustering benchmarks.

Chapter 6 introduces the proposed APP algorithm for modeling lexical semantic change. In this chapter, we first outline the integration of APP into a novel incremental model for lexical semantic change, called What is Done is Done (WiDiD). Then, we illustrate the application of WiDiD in two distinct real-world scenarios spanning multiple time periods. Finally, we assess WiDiD's performance against existing benchmarks for lexical semantic change across two time periods and in various languages.

Chapter 7 outlines several limitations of existing approach comparisons, potentially leading to misleading conclusions in the scientific literature. In this chapter, we first point out the diverse conditions under which existing experiments have been conducted. Then, we systematically evaluate different state-of-the-art LLMs and approaches for modeling lexical semantic change under equal conditions across various language and NLP evaluation tasks. This allows us to establish a reliable comparison of LLMs for modeling LSC.

Chapter 8 introduces a replacement schema to study the effects of lexical semantic change in LLMs. In this chapter, we first investigate the use of lexical replacements derived from lexical resources to analyze LLMs when words undergo semantic change. Then, we propose using lexical replacements and lexical substitutes automatically generated by LLMs to model lexical semantic change.

Chapter 9 investigates the use of automatically generated sense definitions and their utility for modeling word meaning. In this chapter, we first evaluate the use of generative LLMs for generating sense definitions. Then, we propose using sense definitions as intermediate word-meaning representations, subsequently encoded as sentence embeddings to model lexical semantic change.

Chapter 10 proposes a novel evaluation framework for the modeling of historical resonance. In this chapter, we first introduce the novel evaluation framework in relation to existing scientific literature. Then, we evaluate a set of LLMs in modeling historical resonance, operationalized as topical relatedness of text-reuse instances.

Finally, Chapter 11 concludes this thesis with an overall summary and a discussion of the implications of our main contributions.

1.4 Publications

As this thesis was progressing, parts of it were either published as peer-reviewed papers or submitted to prestigious venues. The published papers were presented at ACL-sponsored conferences (i.e., ACL, EACL, NAACL, EMNLP) and their affiliated workshops (i.e., LChange), as well as in scientific journals (i.e., ACM

overview	**RQ1**	**RQ2**	**RQ3**	Publications
Chapter 1	○	○	○	-
Chapter 2	•	○	○	Periti and Montanelli, 2024
Chapter 3	•	○	○	Periti et al., 2024d
Chapter 4	○	•	○	Periti and Tahmasebi, 2024b
Chapter 5	○	•	○	Castano et al., 2024
Chapter 6	○	•	○	Castano et al., 2024; Periti et al., 2024e, 2022
Chapter 7	•	○	○	Periti and Tahmasebi, 2024a
Chapter 8	•	○	○	Periti et al., 2024b
Chapter 9	•	○	○	Periti et al., 2024a
Chapter 10	○	○	•	Periti et al., 2024c
Chapter 11	○	○	○	-

Table 1.1: Overview of the discourse surrounding the defined research questions (RQs) across the entire thesis. For each chapter, we provide the publication references upon which it is based.

Computing Surveys, Language Resources and Evaluation). The paper under review is currently being considered for publication in a computer science journal. Each chapter of this thesis draws partially from one or more of these papers, where we collaborated with other scholars. Therefore, at the beginning of each chapter, we provide a reference directing the reader to the corresponding paper(s). See Table 1.1 for an overview.

To offer a more comprehensive overview, we present here the list of publications upon which this thesis is largely based:

Francesco Periti and Stefano Montanelli. 2024. Lexical Semantic Change through Large Language Models: a Survey. ACM Comput. Surv., 56(11).

Francesco Periti, Haim Dubossarsky, and Nina Tahmasebi. 2024**d**. (Chat)GPT v BERT: Dawn of Justice for Semantic Change Detection. In Findings of the Association for Computational Linguistics: EACL 2024, pages 420–436, St. Julian's, Malta. Association for Computational Linguistics.

Francesco Periti and Nina Tahmasebi. 2024**b**. Towards a Complete Solution to Lexical Semantic Change: an Extension to Multiple Time Periods and Diachronic Word Sense Induction. In Proceedings of the 5th Workshop on Computational Approaches to Historical Language Change, pages 108–119, Bangkok, Thailand. Association for Computational Linguistics.

Silvana Castano, Alfio Ferrara, Stefano Montanelli, and Francesco Periti. 2024. Incremental Affinity Propagation based on Cluster Consolidation and Stratification. eprint 2401.14439, arXiv. Under review.

Francesco Periti, Alfio Ferrara, Stefano Montanelli, and Martin Ruskov. 2022. What is Done is Done: an Incremental Approach to Semantic Shift Detection. In Proceedings of the 3rd Work-

shop on Computational Approaches to Historical Language Change, pages 33–43, Dublin, Ireland. Association for Computational Linguistics.

Francesco Periti, Sergio Picascia, Stefano Montanelli, Alfio Ferrara, and Nina Tahmasebi. 2024**e**. Studying Word Meaning Evolution through Incremental Semantic Shift Detection. Language Resources and Evaluation.

Francesco Periti and Nina Tahmasebi. 2024**a**. A Systematic Comparison of Contextualized Word Embeddings for Lexical Semantic Change. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Pa- pers), pages 4262–4282, Mexico City, Mexico. Association for Computational Linguistics.

Francesco Periti, Pierluigi Cassotti, Haim Dubossarsky, and Nina Tahmasebi. 2024**b**. Analyzing Semantic Change through Lexical Replacements. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4495–4510, Bangkok, Thailand. Association for Computational Linguistics.

Francesco Periti, David Alfter, and Nina Tahmasebi. 2024**a**. Automatically Generated Definitions and their utility for Modeling Word Meaning. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, Florida, USA. Association for Computational Linguistics.

Francesco Periti, Pierluigi Cassotti, Stefano Montanelli, Nina Tahmasebi, and Dominik Schlechtweg. 2024**c**. `TRoTR`: A Framework for Evaluating the Re-contextualization of Text Reuse. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, Florida, USA. Association for Computational Linguistics.

Chapter 2

Modeling lexical semantic change

> "*Innovations which change the lexical meaning rather than the grammatical function of a form, are classed as change of meaning or semantic change*"
>
> Leonard Bloomfield, *Language*

2.1 Introduction

The modeling of lexical semantic change involves the automatic identification, interpretation, and assessment of words that change in meaning over time. Distributional word representations (i.e., word embeddings) generated by LLMs emerged as an effective solution to capture the possible change over time in the meanings of a target word. Any embedding-based approach relies on the well-known distributional hypothesis in Linguistics: "*You shall know a word by the company it keeps*" (Firth, 1957; Harris, 1954) and the foundational premise is that words (and word occurrences) that have similar meanings are encoded closely each other in the embedding space (Chiang and Yogatama, 2023; Mikolov et al., 2013a).

The initial excitement for word embeddings prompted researchers and practitioners to model lexical semantic change by using *static* Language Models (LMs) (Shoemark et al., 2019). These models have been widely adopted and the main approaches have been reviewed in three survey papers (Tahmasebi et al., 2021a; Tang, 2018; Kutuzov et al., 2018). Typically, approaches based on static LMs encode a word into a single semantic embedding, which is then used to detect change in the dominant sense (i.e., word meaning) of the word, without considering its potential additional subordinate senses. However, subordinate senses can change on their own, regardless of their dominant sense. For example, considering the word `rock`, the `music` meaning evolved over time to encompass both `music` and a particular lifestyle, while the `stone` meaning remained unchanged (Hengchen et al., 2021). Thus, the recent introduction of more advanced Transformer architectures (Vaswani et al., 2017) has established the use of LLMs as the preferred tool for modeling semantic change. In contrast with static LMs, approaches based on LLMs typically rely on different word representations according to the context in which a word occurs. For instance, different semantic vectors are generated when the word `rock` in the input sequence is used with the `music` connotation or with the `stone` meaning. This capability facilitates the modeling of linguistic *colexification* phenomena such as homonymy (Sato and Heffernan, 2020) and polysemy (Garí Soler and Apidianaki, 2021). However, although more and more approaches based on LLMs are emerging, a classification framework and a corresponding survey of existing approaches are still missing.

Chapter outline.

This chapter includes materials originally published in the following publication:

> Francesco Periti and Stefano Montanelli. 2024. Lexical Semantic Change through Large Language Models: a Survey. ACM Comput. Surv., 56(11).

In this chapter, we survey the main approaches based on LLMs to model the linguistic phenomenon of lexical semantic change through a corresponding NLP task called Lexical Semantic Change (LSC) (also known as Semantic Shift Detection), emphasizing a computational perspective over a linguistic one. The chapter is organized as follows. In Section 2.2, we define the problem of modeling semantic change using LLMs and outline the related workflow and formalization. In Section 2.3, we present a classification framework based on three dimensions of analysis, namely *meaning representation*, *time-awareness*, and *learning modality*, to effectively describe the featuring properties of both *form*- and *sense*-based approaches in which solutions are typically distinguished. We then discuss the classification of state-of-the-art approaches in Section 2.4. Existing assessment methods and metrics are surveyed to examine how existing approaches measure, interpret, and quantify the semantic change of a word. We provide a comparative analysis of approach performance in Section 2.5. We discuss issues related to the scalability, interpretability, and robustness of computational modeling in Section 2.6. Finally, in Section 2.7, we outline open challenges and relevant considerations.

2.2 Problem statement

Consider a diachronic document corpus $C = \bigcup_{i=1}^{n} C_i$ where C_i denotes a set of documents (e.g., sentences, paragraphs) at time t_i; and a set of target words $\mathcal{W}$ occurring in the corpus C across the entire time span $[t_1, \dots, t_n]$.

Modeling lexical semantic change typically involves:

- *word sense induction*: modeling the meaning(s) of each word $w \in \mathcal{W}$ in each time period $t_1, t_2, \dots, t_n$;
- *semantic change detection*: identifying the words $w \in \mathcal{W}$ that change in meaning across all the contiguous time intervals, namely the pairs of time periods $\langle t_1, t_2 \rangle, \langle t_2, t_3 \rangle, \dots, \langle t_{n-1}, t_n \rangle$.

For the sake of readability, in the following, we consider the LSC problem on a corpus $C = C_1 \cup C_2$ and the change assessment of a given target word $w \in \mathcal{W}$ on a single time interval $\langle t_1, t_2 \rangle$, from time period t_1 to time period t_2. This simplification enables to review the current state-of-the-art in a clear and concise fashion, while being easily extendable to the general case. As a matter of fact, the extension to the whole set of target words $\mathcal{W}$ as well as to all the multiple time periods and contiguous time intervals can be obtained by re-executing a considered approach as many times as needed (Giulianelli et al., 2020). *We will focus on the modeling of LSC over multiple time periods in Chapter 4.*

Different formulations of the problem are possibly depending on various research and assessment questions. The most popular are:

1. **Graded Change Detection**: the goal is to quantify the extent to which a word w change in meaning between C_1 and C_2 (Schlechtweg et al., 2020).
2. **Binary Change Detection**: the goal is to classify a word w as "stable" (without lost or gained senses) or "changed" (with lost or gained senses) between C_1 and C_2 (Schlechtweg et al., 2020).
3. **Sense Gain Detection**: the goal is to recognize whether a word w gained meanings or not between C_1 and C_2 (Zamora-Reina et al., 2022b).
4. **Sense Loss Detection**: the goal is to recognize whether a word w lost meanings or not between C_1 and C_2 (Zamora-Reina et al., 2022b).

2.2.1 The general workflow

The approaches to LSC typically follow the four-step *workflow* presented in Table 2.1. The initial **extraction** stage aims to select all the documents in the corpora containing occurrences (i.e., one or more) of the target word. We refer to these documents as *word usages*. The second **representation** stage has the goal to generate a semantic representation for each word occurrence. An optional **aggregation** stage can be then enforced to group multiple word representations into a single one for detecting similar usages and/or reducing the overall computational complexity. The final **assessment** stage consists in the application of a semantic measure to evaluate how the meanings of the word changed over time.

word usage **extraction**	⟶	word occurrence **representation**	⟶	word vector **aggregation**	⟶	semantic change **assessment**

Table 2.1: A general workflow for modeling lexical semantic change through LLMs.

Word usage extraction. Consider the corpora C_1 and C_2 and the target word w. The goal of this stage is to extract all the contextual usages of w from C_1 and C_2. As the word meanings are influenced by morphology and syntax (Wysocki and Jenkins, 1987), the extraction has to capture the occurrences of w in all its linguistic forms (e.g., singular/plural and gender forms, different verb tenses). For instance, a word may change in meaning only in one of its forms. An example is the Italian word `lucciola` that was historically used with a euphemism for `prostitute`, a meaning that has now become obsolete. Nonetheless, the plural form `lucciole` has consistently retained the more stable sense of `fireflies` (Kutuzov et al., 2021a).

Word occurrence representation. The goal of this stage is to generate a word representation for each occurrence of the word w in C_1 and C_2. Ideally, the word representations of w should be similar for semantically similar word occurrences (i.e., usages) across different documents. A LLM is used to represent each occurrence according to its context. Different types of representations can be used. Possible options are:

- **word embeddings**: a semantic vector in a multi-dimensional space that is directly generated by the Encoder of LLMs, such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), or ELMo (Peters et al., 2018).

- **lexical substitutes**: a bag of words that is generated by a Masked LLM such as BERT and RoBERTa to substitute a specific occurrence of w in a document (Card, 2023). These substitutes are supposed to replace a word without introducing grammatical errors or significantly changing its meaning. For example, suitable substitutes for the word `fly` in the sentence `a noisy fly sat on my shoulder` are `bug`, `beetle`, or `butterfly`; while suitable substitutes in the sentence `we will fly to London` are `walk`, `run`, or `bike` (Kudisov and Arefyev, 2022). Alternatively, Causal LLMs such as GPT (Brown et al., 2020) and LLaMA (Touvron et al., 2023a) can be prompted to generate the substitutes (Periti et al., 2024b; Baez and Saggion, 2023). A **word embedding** vector for each occurrence of w can be computed over the substitutes (i.e., **bag-of-substitutes**) using measures like Term Frequency-Inverse Document Frequency (Tf-Idf).

- **sense definitions**: a descriptive interpretation that is generated by a Causal LLM to represent the occurrence of the word w in a particular document (Giulianelli et al., 2023). For example, an occurrence of the word `bank` may correspond to the definition of `a financial institution`, while another occurrence may correspond to `the edge of a river`. Alternatively, when available, lexical resources like WordNet (Miller, 1994) can be leveraged to obtain sense definitions. Sense definitions can be further processed by the Encoder of LLMs to generate less noisy **sense embedding** representations (Kong et al., 2022), or by Natural Language Generation (NLG) metrics such as BLEU, NIST, ROUGE-L, METEOR, or MoverScore (Huang et al., 2021).

Currently, at the time of this thesis, *contextualized* word embeddings are the most widespread tool in LSC, with very few approaches using the other representations. Thus, we will use word embeddings as a reference for *word occurrence representation*. In the following, we denote the representation of the word w in the i-th document of a corpus C_j as $e_{j,i}$, where $j \in 1, 2$. Then, the representation of the word w in a corpus C_j is defined as: $\Phi_j = \{e_{j,1}, \dots, e_{j,z}\}$, with z being the cardinality of C_j, namely the number of documents in C_j containing w. Finally, the sets of representation vectors generated for the word w at time t_1 and t_2 are denoted as Φ_1 and Φ_2, respectively. *We will focus on the use of lexical substitutes and sense definitions in Chapter 8 and Chapter 9, respectively.*

Word vector aggregation. This stage is optionally executed and it has two main goals: **i)** to recognize when different word occurrences convey a similar meaning, and **ii)** to reduce the number of elements to consider for change detection. To this end, clustering and averaging techniques are proposed for aggregating the generated word embeddings.

i) **Clustering** techniques are employed to group similar word embeddings in a cluster, each one loosely denoting a specific word meaning. In some approaches, it is assumed that the corpus is *static*, meaning that all the documents in C_1 and C_2 are available as a whole. Then, a *joint* clustering operation is executed over the embeddings of $\Phi_1 \cup \Phi_2$ (e.g., Martinc et al., 2020b). In other approaches, it is assumed that the corpus is *dynamic*, meaning that documents become available at different time periods and a *separate* clustering operation is performed over the embeddings of Φ_1 and Φ_2, individually (i.e., one exclusively on Φ_1 and another exclusively on Φ_2 embeddings). When a separate clustering is executed, the resulting clusters need to be aligned in order to recognize similar word meanings at different consecutive time periods (e.g., Kanjirangat et al., 2020). To overcome the need for aligning clusters, an *incremental* clustering operation is employed to progressively group the embedding available at the different time steps (e.g., Periti et al., 2024e). The result of clustering is a set of k clusters where the i-th cluster is denoted as ϕ_i and it can fall into one of the following cases (see Figure 2.1):

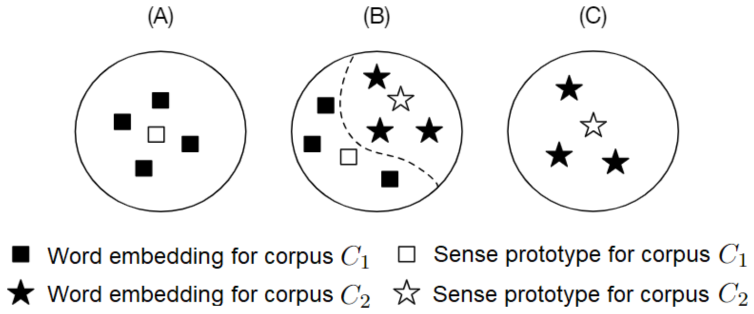

(A)
(B)
(C)
■ Word embedding for corpus C_1
□ Sense prototype for corpus C_1
★ Word embedding for corpus C_2
☆ Sense prototype for corpus C_2

Figure 2.1: Possible cluster composition for modeling word senses over time.

- **(A)**: ϕ_i contains only embeddings from C_1;
- **(B)**: ϕ_i contains a mixture of embeddings from both C_1 and C_2;
- **(C)**: ϕ_i contains only embeddings from C_2.

As a result, a cluster $\phi_i = \phi_{1,i} \cup \phi_{2,i}$ is composed by the union of two partitions $\phi_{1,i}$ and $\phi_{2,i}$ denoting the embeddings from Φ_1 and Φ_2, respectively. When a *joint* or *incremental* clustering is applied, the resulting clusters can belong to any of the above cases (i.e., A, B, and C). When a *separate* clustering is applied, the resulting clusters can just belong to A and C cases, meaning that $\phi_{2,i} = \emptyset$ and $\phi_{1,i} = \emptyset$, respectively.

ii) **Averaging** techniques consist in determining a *prototypical* representation of the word w. As an option, a *word*-prototype can be computed by averaging all its embedding. In this case, *word*-prototypes μ_1 and μ_2 are created as the average embeddings of Φ_1 and Φ_2, respectively (e.g., Kutuzov and Giulianelli, 2020). As an alternative option, averaging can be executed on top of the results of clustering. For each cluster, averaging is used to create a prototypical representation of all the cluster elements (i.e., the centroid of the cluster). In particular, *sense*-prototypes $c_{1,i}$, $c_{2,i}$ can be created for each cluster ϕ_i as the average embedding of its cluster partitions $\phi_{1,i}, \phi_{2,i}$, respectively (e.g., Periti et al., 2022).

Semantic change assessment. This stage has the goal to measure the change on the meanings of the word w across the corpora C_1 and C_2 by considering the sets Φ_1 and Φ_2. In the literature, a number of functions are proposed for semantic change assessment. Distinctions can be made between measures that assess the change by considering the whole set of embedding representations Φ_i, by those that exploit the prototypical representations c_i and/or μ_i generated during the aggregation step through clustering and/or averaging. According to Kutuzov et al. (2018), the definition of a rigorous, formal, mathematical model for representing the assessment functions used in LSC approaches is a challenging issue. In the following, we provide a formal definition of an abstract function f, with the goal of encompassing all existing assessment measures.

The semantic change assessment $s = f(\cdot,\cdot,\cdot)$ is defined as follows:

$$f : \{\mathbb{R}^D\}^{(p_1+z_1\cdot\delta)}, \{\mathbb{R}^D\}^{(p_2+z_2\cdot\delta)}, \ c \rightarrow \mathcal{S}$$

where D is the dimension of the word vectors in Φ_1 and Φ_2; p_1, p_2 are the number of prototypical embeddings under consideration for C_1, C_2, respectively; z_1, z_2 are the number of vectors in Φ_1 and Φ_2, respectively; $\delta \in \{0, 1\}$ is a flag that allows to distinguish the approaches according to the kind of embedding used (i.e., original and/or prototypical); c is a counting function that determines the normalized number of embeddings in the cluster partitions $\phi_{1,i}$ and $\phi_{2,i}$, respectively.

The counting function c is defined as:

$$c : \{\mathbb{R}^D\}^{z_1}, \{\mathbb{R}^D\}^{z_2} \rightarrow \mathbb{R}^k, \mathbb{R}^k$$

where k denotes the comprehensive number of k clusters obtained when a clustering operation is enforced during the aggregation stage. If a cluster ϕ_i contains embeddings only from Φ_1, then the corresponding count for C_2 will be equal to 0, and vice versa. When the clustering operation is not enforced, each embedding is mapped to a singleton group (i.e., $k = z_1 + z_2$).

The signature of f depends on the possible execution of an aggregation technique:

- *Clustering*. When the clustering operation is executed, then $p_1 = p_2 = 0$ and $\delta = 1$. This means that all the $z_1 + z_2$ embeddings in $\Phi_1 \cup \Phi_2$ are exploited for semantic change assessment (e.g., Martinc et al., 2020b).

- *Averaging*. When the averaging operation is executed, then $p_1 = p_2 = 1$. In some approaches, $\delta = 0$ and this means that the function f is defined as a distance measure over prototypical representations (e.g., Martinc et al., 2020a). In some other approaches, $\delta = 1$ and this means that f is defined as a distance measure over the original embeddings Φ and their prototypical representations (e.g., Pömsl and Lyapin, 2020).

- *Clustering + Averaging*. When both clustering and averaging are performed, $p_1, p_2 > 0$ and δ can be both 0 or 1 as in the previous case (e.g., Castano et al., 2024).

The output $\mathcal{S}$, is generally defined according to the formulation of the LSC problem.

- *Graded Change Detection*: $\mathcal{S} = \mathbb{R}$, with s quantifying the change of w between C_1 and C_2.

- *Binary Change Detection*: $\mathcal{S} = \{0, 1\}$, with s representing a binary score for "stable" (i.e., 0) and "changed" (i.e., 1), respectively.

- *Sense Gain Detection*: $\mathcal{S} = \{0, 1\}$, with s representing a binary score for not-gained (i.e., 0) and gained (i.e., 1), respectively.

- *Sense Loss Detection*: $\mathcal{S} = \{0, 1\}$, with s representing a binary score for not-lost (i.e., 0) and lost (i.e., 1), respectively.

Graded Change Detection is the most commonly considered formulation. Thus, in this chapter, we focus on approaches that address LSC considering Graded Change Detection. It is worth noting that conceptually Binary Change Detection is not the binarization of Graded Change Detection. Indeed, even if a word does not gain/lose meanings (i.e., "stable" word), it can be associated with a high value of s due to other forms of semantic change, such as amelioration (change to positive connotation) and pejoration (change to negative connotation) (Goworek and Dubossarsky, 2024). However, in practice, Binary Change Detection is derived from Graded Change Detection by binarizing the graded s through a threshold θ (e.g., Zhou and Li, 2020). We do not address Sense Gain and Sense Loss Detection as they are relatively novel formulations.

For the sake of clarity, a summary of the notation used throughout this chapter is provided in Table 2.2.

2.3 An original classification framework

A consolidated and widely-accepted classification framework of approaches is not available. A basic framework is focused on the meaning representation of the words by distinguishing between *form*- and *sense*-based

Notation	Definition
$\mathcal{C}$	Diachronic document corpus
t_j	Time period j-th
w	Target word
C_j	Set of documents at time t_j containing a word w
$\mathcal{W}$	Set of target words
$e_{j,i}$	Representation (i.e., embedding) of the word w in the i-th document of a corpus C_j
Φ_j	Set of the representations of w in the corpus C_j
ϕ_i	i-th cluster containing the representations of the word w
$\phi_{j,i}$	Subset of representations Φ_j in the cluster ϕ_i
μ_j	Prototypical representation of w for Φ_j
$c_{j,i}$	Prototypical representation of w for $\phi_{j,i}$

Table 2.2: Summary of notation used in this chapter.

approaches (Giulianelli et al., 2020; Qiu and Yang, 2022). However, such a distinction is not universally recognized with a unique interpretation. Sometimes, these two categories are referred as *type*- and *token*-based, where averaging and clustering are enforced to aggregate embeddings, respectively (Laicher et al., 2020; Schlechtweg et al., 2020). More recently, *average*- and *cluster*-based categories have been proposed to rename form and sense ones to highlight the method used for embedding aggregation (Periti et al., 2022).

In the following, we propose a comprehensive classification framework that extends the basic distinction between form- and sense-based approaches by introducing three dimensions of analysis, namely *meaning representation*, *time-awareness*, and *learning modality* (see Table 2.3).

Meaning representation	Time-awareness	Learning modality
form-based sense-based	time-oblivious time-aware	supervised unsupervised

Table 2.3: A classification framework for modeling lexical semantic change.

Meaning representation. Borrowing the distinction proposed by Giulianelli et al. (2020), this dimension focuses on the meaning representation of a word. Two categories are defined:

- *form-based*: the meaning representation concerns the high-level properties of the target word w, such as its degree of polysemy or its dominant sense. When the polysemy is considered, the employed approaches do not enforce any aggregation stage and the semantic change of w is assessed by measuring the degree of change on the embeddings Φ_1 and Φ_2 (i.e., change on the degree of polysemy). When the dominant sense is considered, all the meanings of w are collapsed into a single one on which the change is assessed. Typically, the embeddings Φ_1 and Φ_2 are averaged into corresponding word prototypes μ_1 and μ_2, respectively. In this case, the approaches focus on one meaning of w that can be considered as an approximation of the *dominant sense* since, generally, it is the most frequent in the corpus, and thus the one most represented in the word prototype. We stress that form-based approaches are not able to represent how minor meanings *compete* and *cooperate* to change the dominant sense (Hu et al., 2019).

- *sense-based*: the meaning representation concerns the low-level properties of the target word w, such as its different context usages (i.e., its multiple meanings). All the senses of a word w are represented and considered in the change assessment, namely the dominant sense and the minor ones. Typically, the embeddings Φ_1 and Φ_2 are aggregated into clusters, each one loosely representing a different meaning of w. Sense-based approaches allow to capture the changes over the different meanings of w as well as to interpret the word change (e.g., a new/existing meaning has gained/lost importance).

Time awareness. This dimension focuses on how the time information of the documents is considered by the employed LLM. Two categories are defined:

- *time-oblivious*: this category is based on the assumption that a document of time t adopts linguistic patterns that are known by the LLM and already characterize the language at the time t by its own. Thus, it is not needed that the LLM is aware of the time in which a document is inserted in the corpus. A time-oblivious approach is based on *the contextual nature of embeddings generated by the model, which by definition are dependent on the context that is always time-specific* (Martinc et al., 2020b).

- *time-aware*: this category is based on the assumption that the LLM is not capable of *adapting to time and generalizing temporally* since they are *usually pre-trained on corpora derived from a snapshot of the web crawled at a specific moment in time* (Rosin et al., 2022). Thus, it is needed that the LLM is aware of the time in which a document is inserted in the corpus. As a result, a time-aware LLM encodes the time information as well as the linguistic context of a document while generating the word representations.

Learning modality. This dimension is about the possible use of external knowledge for describing and learning the word meanings to recognize. Two categories are defined:

- *supervised*: a form of supervision is enforced to inject external knowledge to support the change assessment. In addition to the text in the corpora C_1 and C_2, a lexicographic/manual supervision is employed. Lexicographic supervision refers to the use of dictionaries, vocabularies, or thesauri to support word sense induction and recognize the meaning of each word occurrence. This solution can be considered as an alternative to aggregation by clustering for meaning identification. Manual supervision involves using a human-annotated dataset (e.g., Word-in-Context dataset) with gold labels for training or fine-tuning the LLM (Arefyev et al., 2021).

- *unsupervised*: the change assessment is exclusively based on the text of the corpora C_1, C_2 without any external knowledge support. As a result, the word meanings to recognize emerge from the corpora and the change is completely assessed by exploiting unsupervised learning techniques. The use of aggregation by clustering is an example of unsupervised learning for meaning detection.

2.4 A comprehensive review of the state-of-the-art

In this section, the existing approaches in literature are reviewed according to the classification framework discussed in Section 2.3. In particular, the solutions are presented in Sections 2.4.1 and 2.4.2 according to the meaning representation of the considered target word, namely *form-* and *sense-* based approaches, respectively. Moreover, Section 2.4.3 describes the so-called *ensemble* approaches, namely approaches that are based on a combination of multiple form- and/or sense-based solutions.

For the sake of comparison, in each category (i.e., form, sense, ensemble), a summary table is provided to frame the literature papers according to the classification framework as well as to report additional descriptive features about the following aspects:

- *LLM*: the large language model used (e.g., BERT);
- *Training language*: the language of the dataset used for training the model. The possible options are *monolingual* to denote when training is executed on a single language, or *multilingual* when more than one language is considered.
- *Type of training*: how the model is trained. Five categories are distinguished:
  - *trained*: the model is trained from scratch through a typical objective function(s);
  - *pre-trained*: the model has been pre-trained on a large dataset by other researchers, and it is directly used as an off-the-shelf solution instead of being trained from scratch;
  - fine-tuned for *domain-adaptation*: the model has been pre-trained on a large dataset by other researchers, then it is fine-tuned on new data through the same objective function;
  - fine-tuned for *incremental domain-adaptation*: the model is fine-tuned on the corpus of the first time period C_1. Then, it is re-tuned separately on the corpus C_2. The model at time t_2 is initialized with the weights from the model at time t_1, so that both models are inherently related the one to the other;
  - *fine-tuned*: the model has been pre-trained on a large dataset by other researchers, then it is fine-tuned on new data through a different objective function.
- *Layer*: the architecture's layer(s) from which word representations are extracted;
- *Layer aggregation*: the type of aggregation used to synthesize the word representations extracted from different layers into a single embedding;
- *Clustering algorithm*: the clustering algorithm used in the aggregation stage;
- *Change function*: the function f used to detect/assess the semantic change;
- *Corpus language*: the natural language of the corpus in the considered experiments of change assessment (e.g., English, Italian, Spanish).

2.4.1 Form-based approaches

Ref.	**Time awareness**	**Learning modality**	**LLM**	**Training language**	**Type of training**	**Layer**	**Layer aggregation**	**Clustering algorithm**	**Change function**	**Corpus language**
Arefyev et al. 2021	time-oblivious	supervised	XLM-R-large	multilingual	fine-tuned	last	-	-	APD	Russian
Beck 2020	time-oblivious	unsupervised	mBERT-base	multilingual	pre-trained	last two	average	K-Means	CD	English, German, Latin, Swedish
Martinc et al. 2020a	time-oblivious	unsupervised	BERT-base, mBERT-base	monolingual, multilingual	domain-adaptation	last four	sum	-	CD	English, Slovenian
Horn 2021	time-oblivious	unsupervised	BERT-base, RoBERTa-base	monolingual	domain-adaptation, pre-trained	-	-	-	CD	English
Hofmann et al. 2021	time-aware	unsupervised	BERT-base	monolingual	fine-tuned	last	-	-	CD	English
Zhou and Li 2020	time-aware	unsupervised	BERT-base	monolingual	domain-adaptation	last four	sum	-	CD	English, German, Latin, Swedish
Rosin et al. 2022	time-aware	unsupervised	BERT-base, BERT-tiny	monolingual	fine-tuned	all, last, last four	average	-	CD, TD	English, Latin
Rosin and Radinsky 2022	time-aware	unsupervised	BERT-base, BERT-small, BERT-tiny	monolingual	fine-tuned	all, last, last four, last two	average	-	CD	English, German, Latin
Kutuzov and Giulianelli 2020	time-oblivious	unsupervised	BERT-base, ELMo, mBERT-base	monolingual, multilingual	domain-adaptation, incremental domain-adaptation, pre-trained, trained	all, last, last four	average	-	APD, CD, PRT	English, German, Latin, Swedish
Giulianelli et al. 2020	time-oblivious	unsupervised	BERT-base	monolingual	pre-trained	all	sum	-	APD	English
Keidar et al. 2022	time-oblivious	unsupervised	RoBERTa-base	monolingual	domain-adaptation	all, first, last	sum	-	APD	English
Pömsl and Lyapin 2020	time-aware	unsupervised	BERT-base, mBERT-base	monolingual, multilingual	fine-tuned	last	-	-	APD	English, German, Latin, Swedish
Kudisov and Arefyev 2022	time-oblivious	unsupervised	XLM-R-large	multilingual	pre-trained	-	-	-	APD	Spanish
Laicher et al. 2021	time-oblivious	unsupervised	BERT-base	monolingual	pre-trained	first, first + last, first four, last, last four	average	-	APD, APD-OLD/NEW, CD	English, German, Swedish
Wang et al. 2020	time-oblivious	unsupervised	mBERT-base	multilingual	pre-trained	last	-	-	APD, HD	Italian
Kutuzov 2020	time-oblivious	unsupervised	BERT-base, BERT-large, ELMo, mBERT-base	monolingual, multilingual	domain-adaptation, pre-trained	all, last, last four	average	-	APD, DIV, PRT	English, German, Latin, Swedish, Russian
Ryzhova et al.	time-oblivious	unsupervised	ELMo, **RuBERT** Kuratov and Arkhipov 2019	multilingual	pre-trained, trained	-	-	-	APD	Russian
Rodina et al. 2021	time-oblivious	unsupervised	ELMo, RuBERT	monolingual, multilingual	domain-adaptation	last	-	-	PRT	Russian
Liu et al. 2021b	time-oblivious	unsupervised	BERT-base, **LatinBERT** Bamman and J. Burns 2020	multilingual, monolingual	domain-adaptation	last four	sum	-	CD	English, German, Latin, Swedish
Giulianelli et al. 2022	time-oblivious	unsupervised	XLM-R-base	multilingual	domain-adaptation	all	average	-	APD, PRT	English, German, Italian, Latin, Norwegian, Russian, Swedish
Laicher et al. 2020	time-oblivious	unsupervised	mBERT-base	multilingual	pre-trained	all, last four	average	-	APD	Italian
Qiu and Yang 2022	time-oblivious	unsupervised	BERT-base	monolingual	domain-adaptation pre-trained	last four	sum	-	CD	English
Periti et al. 2022	time-oblivious	unsupervised	BERT-base mBERT-base	monolingual, multilingual	pre-trained	last four	sum	-	CD, DIV	English, Latin
Montariol et al. 2021	time-oblivious	unsupervised	BERT-base mBERT-base	monolingual, multilingual	domain-adaptation	last four	sum	-	CD	English, German, Latin, Swedish

Table 2.4: Summary view of form-based approaches. Missing information is denoted with a dash.

According to Table 2.4, we note that most form-based approaches are time-oblivious. A few time-aware approaches have been recently appeared and they are all characterized by the adoption of a specific fine-tuning operation to inject time information into the model. All the current work leverage unsupervised learning modalities with the exception of Arefyev et al. (2021). The aggregation stage is mostly based on averaging, while clustering is only enforced by Beck (2020) where a cluster represents the dominant sense of the word w. In particular, Beck (2020) consider a word as changing when clustering the embeddings Φ_1 and Φ_2 via K-means with $k = 2$ generates two groups where one of the two clusters contains at least 90% of the embeddings from one corpus only (i.e., C_1 or C_2).

In form-based approaches, the following change functions are proposed for measuring the semantic change s.

Cosine distance (CD). The change s is measured as the *cosine distance* (CD) between the word prototypes μ_1, μ_2 as follows:

$$CD(\mu_1, \mu_2) = 1 - CS(\mu_1, \mu_2) \tag{2.1}$$

where CS is the *cosine similarity* between the prototypes. Intuitively, the greater the $CD(\mu_1, \mu_2)$, the greater the change in the dominant sense of w.

Typically, the prototypes μ_1 and μ_2 are determined through aggregation by averaging over Φ_1 and Φ_2, respectively (e.g., Martinc et al., 2020a). As a difference, Horn (2021) compute the prototype embedding μ_2 at time step $t = 2$ by updating the prototype embedding μ_1 at time step $t = 1$ through a weighted running average (e.g., Finch, 2009).

Martinc et al. (2020a) employ the CD metric in a multilingual experiment where the change is measured across a diachronic corpus with texts of different languages. This is the only example of cross-language change detection.

CD is also used in time-aware approaches. The integration of extra-linguistic information into word embeddings, such as time and social space, has been proposed in previous work based on static LMs (Rudolph and Blei, 2018; Zeng et al., 2018). Recently, this integration has been also applied to contextualized embeddings (Huang and Paul, 2019; Röttger and Pierrehumbert, 2021). Hofmann et al. (2021) fine-tune a pre-trained LLM to encapsulate time and social space in the generated embeddings. Then, the change s is assessed by computing the CD between embeddings generated by the original pre-trained model and the embeddings generated by the time-aware, fine-tuned model. In particular, Zhou and Li (2020) adopt a *temporal referencing* mechanism to encode time-awareness into a pre-trained model. Temporal referencing is a pre-processing step of the documents that tags each occurrence of w in C_1 and C_2 with a special marker denoting the corpus/time in which it appears (Ferrari et al., 2017; Dubossarsky et al., 2019). The embeddings of a tagged word are learned by fine-tuning the LLM for domain-adaptation. In this case, s is assessed by computing the CD between $\mu_{[1]}$ and $\mu_{[2]}$, where $[i]$ denotes w with the temporal marker t_i. Similarly to Zhou and Li (2020), a time-aware approach is proposed by Rosin et al. (2022) where a time marker is added to documents instead of words and the LLM is fine-tuned to predict the injected time information (i.e., time masking). This way, there is no need to add a tag for each target word and its various forms (e.g., singular, plural), thereby avoiding the inclusion of additional new tokens in the LLM's vocabulary. As an alternative, Rosin and Radinsky (2022) adopt a *temporal attention* mechanism to generate the embeddings Φ_1 and Φ_2 for calculating CD.

Inverted similarity over word prototype (PRT). This measure is proposed as an alternative to CD for improving the effectiveness of the change detection (Kutuzov and Giulianelli, 2020). The *inverted similarity over word prototypes* (PRT) measure is defined as:

$$PRT(\mu_1, \mu_2) = \frac{1}{CS(\mu_1, \mu_2)} \ . \tag{2.2}$$

Time-diff (TD). This measure is designed for time-aware approaches and it works on analyzing the change of polysemy of a word over time. It is based on the model's capability to predict the time of a document and it calculates the change s by considering the probability distribution of the predicted times (Rosin et al., 2022). Intuitively, a uniform distribution means that the association document-time is not strong enough to clearly entail a change. Instead, a non-uniform distribution means that there is evidence to predict the time of a document. Consider a document d, let $p_j(d)$ be the probability of d to belong to the time t_j. The function *time diff* (TD) is defined as the average difference of the predicted time probabilities:

$$TD(C_1, C_2) = \frac{1}{|C_1 \cup C_2|} \sum_{d_1 \in C_1, d_2 \in C_2} |p_1(d_1) - p_2(d_2)| \ . \tag{2.3}$$

The experiments conducted by Rosin et al. (2022) demonstrate that TD outperforms CD in short-term semantic change when their performance is compared on the task of Graded Change Detection across various benchmarks. On the contrary, CD outperforms TD over long-term semantic change. Rosin et al. (2022) argue that TD is less effective on long-term periods since major differences in writing style emerge and the prediction of document-time associations is less reliable.

Average pairwise distance (APD). This measure exploits the variance of the contextualized representations Φ_1, Φ_2 to compute the semantic change assessment (i.e., variance on the word polysemy). As a difference from the previous measures, APD directly works on word embeddings without requiring any aggregation stage, namely clustering nor averaging. The *average pairwise distance* (APD) is defined as follows:

$$APD(\Phi_1, \Phi_2) = \frac{1}{|\Phi_1||\Phi_2|} \cdot \sum_{e_{1,i} \in \ \Phi_1, \ e_{2,i} \in \ \Phi_2} d(e_{1,i}, e_{2,i}) \ , \tag{2.4}$$

where d is an arbitrary distance measure (e.g., cosine distance, euclidean distance, canberra distance). According to the experiments performed by Giulianelli et al. (2020), APD better performs when the euclidean distance is employed as d. Keidar et al. (2022) use APD over the embeddings Φ_1 and Φ_2 by applying a dimensionality reduction through the Principal Component Analysis (PCA). Experiments on both slang and non-slang words are performed through causal analysis to study how distributional factors (e.g., polysemy, frequency shift) influence the change s. The results show that slang words experience fewer semantic change than non-slang words.

Kudisov and Arefyev (2022) use lexical substitutes to assess s. A set of lexical substitutes is generated by leveraging a masked LLM (e.g., XLM-R) and word representations Φ_1, and Φ_2 are computed as *bag-of-substitutes*. Then, APD is finally computed over Φ_1, and Φ_2 to assess s.

APD is also used in a time-aware approach described by Pömsl and Lyapin (2020), where a pre-trained BERT model is fine-tuned to predict the time period of a sentence. APD is finally used to measure the change between the embeddings extracted from the fine-tuned LLM.

Arefyev et al. (2021) employ APD to measure the change s over the embeddings Φ_1 and Φ_2 extracted from a supervised Word-in-Context model (WiC, Pilehvar and Camacho-Collados, 2019). This LLM is trained to reproduce the behavior of human annotators when they are asked to evaluate the similarity of the meaning of a word w in a pair of given sentences from C_1 and C_2, respectively. The embeddings Φ_1 and Φ_2 are extracted from the trained WiC model for calculating the final APD measure.

Average of average inner distances (APD-OLD/NEW). The APD-OLD/NEW measure is presented by Laicher et al. (2021) as an extension of APD and it estimates the change s as the average degree of polysemy of w in the corpora C_1 and C_2, respectively. The *average of average inner distances* (APD-OLD/NEW) is defined as:

$$APD\text{-}OLD/NEW(\Phi_1, \Phi_2) = \frac{AID(\Phi_1) + AID(\Phi_2)}{2} \, . \tag{2.5}$$

where AID is the *average inner distance* and it measures the degree of polysemy of w in a specific time frame by relying on the APD measure, namely $AID(\Phi_1) = APD(\Phi_1, \Phi_1)$ and $AID(\Phi_2) = APD(\Phi_2, \Phi_2)$, respectively.

Hausdorff distance (HD). The change s is measured as the *Hausdorff distance* (HD) between the word embeddings Φ_1 and Φ_2. Similarly to APD, HD directly works on word embeddings without requiring any aggregation stage. HD relies on the euclidean distance d to measure the difference between the embeddings of w in C_1 and C_2 and it returns the greatest of all the distances d from one embedding $e_1 \in \Phi_1$ to the closest embedding $e_2 \in \Phi_2$, or vice-versa. The HD measure is defined as follows:

$$HD(\Phi_1, \Phi_2) = \max\left(\sup_{e_1 \in \Phi_1} \inf_{e_2 \in \Phi_2} d(e_1, e_2), \sup_{e_2 \in \Phi_2} \inf_{e_1 \in \Phi_1} d(e_2, e_1) \right) \, . \tag{2.6}$$

The experiments performed by Wang et al. (2020) show that HD is sensitive to outliers since it is based on infimum and supremum, thus an outlier embedding may largely affect the final s value.

Difference between token embedding diversities (DIV). Similar to APD, this measure assesses the change s by exploiting the variance of the contextualized representation Φ_1 and Φ_2. As a difference with APD, the *difference between token embedding diversities* (DIV) leverages a coefficient of variation calculated as the average of the cosine distances d between the embeddings Φ_1 and Φ_2, and their prototypical embeddings μ_1 and μ_2, respectively (Kutuzov, 2020). The intuition is that when w is used in just one sense, its embeddings tend to be close to each other yielding a low coefficient of variation. On the opposite, when w is used in many different senses, its embeddings are distant to each other yielding to a high coefficient of variation. DIV is defined as the absolute difference between the coefficient of variation in C_1 and C_2:

$$DIV(\Phi_1, \Phi_2) = \left| \frac{\sum_{e_1 \in \Phi_1} d(e_1, \mu_1)}{|\Phi_1|} - \frac{\sum_{e_2 \in \Phi_2} d(e_2, \mu_2)}{|\Phi_2|} \right| \tag{2.7}$$

The experiments of Kutuzov (2020) show that when the coefficient of variation is low, the prototypical embeddings μ_1 and μ_2 successfully represent the meanings of the given word w. On the opposite, when the coefficient of variation is high, the prototypical embeddings μ_1 and μ_2 do not provide a relevant representation of the w meanings.

2.4.2 Sense-based approaches

According to Table 2.5, we note that all the sense-based approaches are time-oblivious and that fine-tuning is sometimes adopted, but mainly for domain-adaptation purposes. Most papers leverage unsupervised learning modalities. Only a few exceptions employ a lexicographic supervision (i.e., Hu et al., 2019; Rachinskiy and Arefyev, 2021, 2022). As a difference with form-based, sense-based approaches usually enforce clustering in the aggregation stage. The aggregation by averaging is only exploited by Periti et al.; Hu et al.; Montariol et al. (2022; 2019; 2021), where sense prototypes are computed on top of the results of a clustering operation.

When clustering is adopted, the function f that calculates the change s can be directly defined over the embeddings Φ_1 and Φ_2. As an alternative, the function f can be defined over the distribution of the embeddings in the resulting clusters (i.e., *cluster distribution*). In this case, as a result of the clustering operation, a counting function c is used to determine two cluster distributions p_1 and p_2 that represent the normalized number of embeddings in the cluster partitions $\phi_{1,i}$ and $\phi_{2,i}$, respectively (see Section 2.2). The i-th value $p_{j,i}$ in p_j (with $j \in \{1, 2\}$) represents the number of embeddings of $\phi_{j,i}$ in the i-th cluster, namely: $p_{j,i} = \frac{|\phi_{j,i}|}{|\Phi_j|}$. Finally, the function f is defined as a compound function $f = g \circ c$, where the result of the c function is exploited by a change function g which works on the cluster distributions p_1 and p_2.

In sense-based approaches, the following change functions are proposed for measuring the semantic change s.

Maximum novelty score (MNS). This measure exploits the cluster distributions p_1 and p_2 by leveraging the idea that the higher is the ratio between the number of embeddings Φ_1 and Φ_2 in a cluster, the higher is the semantic change of the considered word w. The *maximum novelty score* (MNS) is defined as:

$$MNS(p_1, p_2) = \max\{NS(p_{1,1}, p_{2,1}), ..., NS(p_{1,k}, p_{2,k})\}\ , \tag{2.8}$$

where $NS(p_{1,i}, p_{2,i}) = p_{1,i}/p_{2,i}$ is the *novelty score* proposed by Cook et al. (2014), and k is the number of clusters produced as a result of the aggregation stage.

Hu et al. (2019) employ MNS as a change measure in a supervised learning approach. In particular, a

Ref.	Time awareness	Learning modality	LLM	Training language	Type of training	Layer	Layer aggregation	Clustering algorithm	Change function	Corpus language
Hu et al. 2019	time.-obl.	supervised	BERT-base	monol.	pre-trained	last	-	-	MNS	English
Rachinskiy and Arefyev 2021	time.-obl.	supervised	XLM-R-base	multil.	fine-tuned, pre-trained	-	-	-	APD	Russian
Rachinskiy and Arefyev 2022	time.-obl.	supervised	XLM-R-base	multil.	fine-tuned, pre-trained	last	-	-	APD, JSD	Spanish
Periti et al. 2022	time.-obl.	unsuperv.	BERT-base, mBERT-base	monol., multil.	pre-trained	last four	sum	AP, APP, IAPNA	JSD, PDIS, PDIV	English, Latin
Montariol et al. 2021	time.-obl.	unsuperv.	BERT-base, mBERT-base	monol., multil.	dom.-ada.	last four	sum	K-Means, AP	JSD, WD	English, German, Latin, Swedish
Rodina et al. 2021	time.-obl.	unsuperv.	mBERT-base, ELMo	monol., multil.	dom.-ada.	last	-	K-Means, AP	JSD MS	Russian
Kanjirangat et al. 2020	time.-obl.	unsuperv.	mBERT-base	multil.	pre-trained	last four	concatenation	K-Means	CSC, JSD	English, German, Latin, Swedish
Giulianelli et al. 2020	time.-obl.	unsuperv.	BERT-base	monol.	pre-trained	all	sum	K-Means	ED, JSD	English
Arefyev and Zhikov 2020	time.-obl.	unsuperv.	XLM-R-base	multil.	dom.-ada.	-	-	AGG	CDCD	English, German, Latin, Swedish
Kashleva et al. 2022	time.-obl.	unsuperv.	BERT-base	monol.	dom.-ada.	all	sum	K-Means	APDP	Spanish
Martinc et al. 2020c	time.-obl.	unsuperv.	BERT-base, mBERT-base	monol., multil.	dom.-ada.	last four	sum	K-Means, AP	JSD	English, German, Latin, Swedish
Kutuzov and Giulianelli 2020	time.-obl.	unsuperv.	BERT-base, ELMo, mBERT-base	monol., multil.	dom.-ada., in. dom.-ada., pre-trained	all, last, last four	average	AP	JSD	English, German, Latin, Swedish
Giulianelli et al. 2022	time.-obl.	unsuperv.	XLM-R-base	multil.	dom.-ada.	all	average	AP	JSD	English, German, Italian, Latin, Norwegian, Russian, Swedish
Wang et al. 2020	time.-obl.	unsuperv.	mBERT-base	multil.	dom.-ada.	last	-	GMMs, K-Means	JSD	Italian
Keidar et al. 2022	time.-obl.	unsuperv.	RoBERTa-base	monol.	dom.-ada.	all, first, last	sum	AP, K-Means, GMMs	ED, JSD	English
Karnysheva and Schwarz 2020	time.-obl.	unsuperv.	ELMo, mELMo	monol., multil.	pre-trained	all	-	K-Means, DBSCAN	JSD	English, German, Latin, Swedish
Cuba Gyllensten et al. 2020	time.-obl.	unsuperv.	XLM-R-base	multil.	pre-trained	last	-	K-Means	JSD	English, German, Latin, Swedish
Rother et al. 2020	time.-obl.	unsuperv.	mBERT-base, XLM-R-base	multil.	pre-tuned	last	-	BIRCH, DBSCAN, GMMs, HDBSCAN	JSD	English, German, Latin, Swedish

Table 2.5: Summary view of sense-based approaches. Missing information is denoted with a dash.

lexicographic supervision (i.e., the Oxford English dictionary) is employed to provide the meanings of the target word w. Each word occurrence in Φ_1 and Φ_2 is associated with the closest meaning of the dictionary according to the cosine distance. As a result, for each word/dictionary meaning, a cluster of word embeddings is defined and MNS is exploited to calculate the overall change.

Maximum square (MS). This measure is an alternative to MNS to assess the change of s. The intuition of MS is that slight changes in cluster distributions p_1 and p_2 may occur due to noise and do not represent a real semantic change (Rodina et al., 2021). The *maximum square* (MS) aims at identifying strong changes in the cluster distributions. As a difference with MNS, the square difference between $p_{1,i}$ and $p_{2,i}$ is used to capture the degree of change instead of the novelty score (NS):

$$MS(p_1, p_2) = \max_i \left(p_{1,i} - p_{2,i}\right)^2 \tag{2.9}$$

Jensen-Shannon divergence (JSD). This measure extends the Kullback-Leibler (KL) divergence, which calculates how one probability distribution is different from another. The *Jensen-Shannon divergence* (JSD) calculates the change s as the symmetrical KL score of the cluster distributions p_1 from p_2, namely:

$$JSD(p_1, p_2) = \frac{1}{2}\left(KL(p_1||M) + KL(p_2||M)\right) \ , \tag{2.10}$$

where KL is the Kullback-Leibler divergence and $M = (p_1 + p_2)/2$.

JSD is also used in approaches where aggregation by clustering is performed separately over the embeddings Φ_1 and Φ_2 (Kanjirangat et al., 2020). As a result, the clusters need to be aligned to determine the distributions p_1 and p_2 before the JSD calculation. As a difference with Kanjirangat et al. (2020), an evolutionary clustering algorithm is employed by Periti et al. (2022) to apply the JSD measure without requiring any alignment step over the resulting clusters.

As a final remark, JSD can be employed to measure the change s over more than two time periods. However, the experiments of Giulianelli et al. (2020) show that the JSD effectiveness over a single time period outperforms the version over more time periods since JSD is insensitive to the order of the temporal intervals.

Coefficient of semantic change (CSC). This measure is proposed as an alternative to JSD where the difference over the weighted number of elements in $\phi_{1,i}$ and $\phi_{2,i}$ for each cluster i is employed to replace KL in measuring the change (Kanjirangat et al., 2020). The *coefficient of semantic change* (CSC) is defined as follows:

$$CSC(p_1, p_2) = \frac{1}{P_1 \cdot P_2} \sum_{k=1}^{K} |P_2 \cdot p_{1,k} - P_1 \cdot p_{2,k}| \ , \tag{2.11}$$

where $P_j = \sum_{i=1}^{k} p_{j,i}$ is the weight of each cluster distribution and k is the number of clusters.

Cosine distance between cluster distributions (CDCD). As a further alternative of JSD, this measure assesses the change s by considering the cluster distributions p_1 and p_2 as vectors and by applying the cosine distance over them to assess the semantic change s. The *cosine distance between cluster distributions* (CDCD) is defined as follows:

$$CDCD(p_1, p_2) = 1 - \frac{p_1 \cdot p_2}{\|p_1\| \times \|p_2\|} \tag{2.12}$$

In Arefyev and Zhikov (2020), CDCD is calculated between the cluster distributions p_1 and p_2 obtained by enforcing clustering over bag-of-substitutes (see the description of Arefyev and Zhikov, 2020 in Section 2.4.1).

Entropy difference (ED). This measure is based on the idea that the higher is the uncertainty in the interpretation of a word occurrence due to the w polysemy in C_1 and C_2, the higher is the semantic change s. The intuition is that high values of ED are associated with the broadening of a word's interpretation, while negative values indicate a narrowing interpretation (Giulianelli et al., 2020). The *entropy difference* (ED) is defined as follows:

$$ED(p_1, p_2) = \eta(p_1) - \eta(p_2) \, , \tag{2.13}$$

where $\eta(p_j)$ is the degree of polysemy of w in the corpus C_j, which is calculated as the normalized entropy of its cluster distribution p_j:

$$\eta(p_j) = \log_K \left(\prod_{k=1}^{K} {p_{j,i}}^{-p_{j,i}} \right) \, .$$

As shown by Giulianelli et al. (2020), ED is not capable of properly assessing s when new usage types of w emerge, while old ones become obsolescent at the same time, since it may lead to no entropy reduction.

Cosine distance between semantic prototypes (PDIS). This measure is presented by Periti et al. (2022) as an extension of the CD measure adopted by form-based approaches. The idea of PDIS is that the aggregation by averaging over cluster prototypes can be employed to produce summary descriptions of the cluster contents (i.e., *semantic prototypes*). The *cosine distance between semantic prototypes* (PDIS) is defined as the CD between $\bar{c}_1$, $\bar{c}_2$, that is:

$$PDIS(\bar{c}_1, \bar{c}_2) = 1 - \frac{\bar{c}_1 \cdot \bar{c}_2}{\|\bar{c}_1\| \times \|\bar{c}_2\|} \tag{2.14}$$

where $\bar{c}_1$ and $\bar{c}_2$ are semantic prototypes defined as the average embeddings of all the sense prototypes $c_{1,i}$ and $c_{2,i}$, respectively.

Difference between prototype embedding diversities (PDIV). This measure is presented by Periti et al. (2022) as an extension of the DIV measure adopted by form-based approaches. PDIV leverages the same intuition of PDIS, namely the semantic prototypes can be employed to calculate the coefficient of ambiguity of w by measuring the difference between a semantic prototype $\bar{c}_j$ and each sense prototype $c_{j,i}$. The *difference between prototype embedding diversities* (PDIV) is defined as the absolute difference between these ambiguity coefficients:

$$PDIV(\Psi_1, \Psi_2) = \left| \frac{\sum_{c_{1,k} \in \Psi_1} d(c_{1,k}, \bar{c}_1)}{|\Psi_1|} - \frac{\sum_{c_{2,k} \in \Psi_2} d(c_{2,k}, \bar{c}_2)}{|\Psi_2|} \right| \, , \tag{2.15}$$

where Ψ_1 and Ψ_2 denote the set of sense prototypes of $c_{1,i}$ and $c_{2,i}$, respectively.

Average pairwise distance (APD). In addition to form-based approaches (see Section 2.4.1), the APD measure is exploited to assess s also in sense-based approaches. Rachinskiy and Arefyev; Rachinskiy and Arefyev (2021; 2022) apply APD to the contextualized embeddings Φ_1 and Φ_2 extracted from a fine-tuned XLM-R model. In particular, an English corpus is used to fine-tune the pre-trained LLM to select the most appropriate WordNet's definition for each word occurrence (Blevins and Zettlemoyer, 2020). As a result of the fine-tuning, both WordNet's definitions and word occurrences are embedded in the same vector space and the meaning of any word occurrence can be induced by selecting the closest definition in the vector space. In Rachinskiy and Arefyev (2021), the zero-shot, cross-lingual transferability property of XLM-R is exploited to obtain word representations for the Russian language and APD is finally applied (Chang et al., 2008; Choi et al., 2021). Rachinskiy and Arefyev (2021) claim that the approach is useful to overstep the lack of lexicographic supervision for low-resource languages and that most concept definitions in English also hold in other languages, such as Russian. However, this claim is not completely satisfied, since some words can drastically change their meaning across languages. For example, the Russian word "пионер" (i.e., pioneer, scout) is strongly connected to the Communist ideology in the Soviet Period, but it isn't in the English language.

Average pairwise distance between sense prototypes (APDP). This measure is an extension of APD and it considers all the pairs of sense prototypes $c_{1,i}$ and $c_{2,i}$ instead of all the original embeddings in Φ_1 and Φ_2 (Kashleva et al., 2022). The *average pairwise distance between sense prototypes (APDP)* is defined as:

$$APD(\Psi_1, \Psi_2) = \frac{1}{|\Psi_1||\Psi_2|} \cdot \sum_{c_{1,k} \in \Psi_1,\ c_{2,k} \in \Psi_2} d(c_{1,k}, c_{2,k}) \tag{2.16}$$

Wassertein distance (WD). This measure models the change assessment as an *optimal transport problem* and it is exploited as an alternative to cluster alignment when aggregation by clustering is performed separately over the embeddings Φ_1 and Φ_2 (Montariol et al., 2021). WD quantifies the effort of re-configuring the cluster distribution of p_1 into p_2, namely minimizing the cost of moving one unit of mass (i.e., a sense prototype) from Ψ_1 to Ψ_2. The *Wassertein distance* (WD) is defined as:

$$WD(p_1, p_2) = \min_{\gamma} \sum_{i}^{k_1} \sum_{j}^{k_2} CD(c_{1,i}, c_{2,j})\, \gamma_{c_{1,i} \rightarrow c_{2,j}} \tag{2.17}$$

$$\text{such that:} \quad \gamma_{c_{1,i} \rightarrow c_{2,j}} \geq 0$$

$$\sum_{i} \gamma_{c_{1,i} \rightarrow c_{2,j}} = p_1$$

$$\sum_{j} \gamma_{c_{1,i} \rightarrow c_{2,j}} = p_2$$

where all $\gamma_{c_{1,i} \rightarrow c_{2,j}}$ represents the (unknown) effort required to reconfigure the mass distribution p_1 into p_2; k_1 and k_2 are the number of clusters obtained by clustering Φ_1 and Φ_2, respectively; CD is the cosine distance computed over the sense prototypes $c_{1,i} \in \Psi_1$ and $c_{2,j} \in \Psi_2$ (Bonneel et al., 2011).

2.4.3 Ensemble-based approaches

In this section, we review the approaches that rely on an *ensemble mechanism*, namely the combination of two or more assessment functions to determine the semantic change score. Ensembling can mean that more than one form- and/or sense-based measure is adopted in a given approach. Ensembling can also mean that a disciplined use of both static and large LMs is used. A final semantic change score is then returned by the whole ensemble process.

Ref.	Time awareness	Learning modality	Language model	Training language	Type of training	Layer	Layer aggregation	Clustering algorithm	Change function	Corpus language
Pömsl and Lyapin 2020	time-aware	unsupervised	BERT-base, mBERT-base	monolingual, multilingual	fine-tuned	last	-	-	APD	English, German, Latin, Swedish
Teodorescu et al. 2022	time-oblivious	unsupervised	XLM-large	multilingual	trained	last four	sum	-	APD	Spanish
Martinc et al. 2020c	time-oblivious	unsupervised	BERT-base, mBERT-base	monolingual, multilingual	domain-adaptation	last four	sum	AP	CD, JSD	English, German, Latin, Swedish
Wang et al. 2020	time-oblivious	unsupervised	mBERT-base	multilingual	pre-trained	last	-	GMMs, K-Means	APD, HD, JSD	Italian
Giulianelli et al. 2022	time-oblivious	unsupervised	XLM-R-base	multilingual	domain-adaptation	all	average	-	APD, PRT	English, German, Italian, Latin, Norwegian, Russian, Swedish
Ryzhova et al. 2021	time-oblivious	unsupervised	ELMo, RuBERT	monolingual, multilingual	pre-trained trained	-	-	-	APD	Russian
Kutuzov et al. 2022b	time-oblivious	unsupervised	BERT-base, ELMo	monolingual, multilingual	domain adaptation	last	-	-	APD, PRT	English, German, Latin, Swedish
Rachinskiy and Arefyev 2021	time-oblivious	supervised	XLM-R-base	multilingual	fine-tuned, pre-trained	-	-	-	APD	Russian
Rosin and Radinsky 2022	time-aware	unsupervised	BERT-base	monolingual	fine-tuned	-	-	-	CD	English, Latin, German

Table 2.6: Summary view of ensemble approaches. Missing information is denoted with a dash.

According to Table 2.6, we note that all the ensemble approaches are time-oblivious with the exception of Pömsl and Lyapin (2020) and Rosin and Radinsky (2022). We also note that unsupervised learning modalities are adopted with the exception of Rachinskiy and Arefyev (2021). As a further remark, most of the ensemble solutions exploit LLMs trained over different languages.

Some ensemble approaches combine form-based and sense-based measures to improve the quality of results. On the one hand, form-based measures are exploited to better capture the dominant sense of the target word w. On the other hand, sense-based measures are exploited to represent all the meanings of w, including the minor ones. The combination of CD (see form-based approaches in Section 2.4.1) and JSD (see sense-based approaches in Section 2.4.2) is proposed by Martinc et al. (2020c). As a further ensemble experiment, the results of combining APD, HD, and JSD are discussed by Wang et al. (2020). The APD measure is also considered by Rachinskiy and Arefyev (2021), where multiple change scores are calculated by using different distance metrics (e.g., Manatthan distance, CD, euclidean distance) and these scores are exploited to train a regression model as an ensemble.

Ensemble approaches based on two form-based measures are also proposed. For instance, Giulianelli et al. (2022) obtain the final semantic change s by averaging APD and PRT scores. This is motivated by experimental results where sometimes APD outperforms PRT, while some other times PRT outperforms APD (Kutuzov and Giulianelli, 2020).

Some other ensemble approaches are based on the idea to combine static and contextualized embeddings. The intuition is that static embeddings can capture the dominant sense of the target word w, better than form-based, contextualized embeddings. In Pömsl and Lyapin; Teodorescu et al. (2020; 2022), the semantic change s is assessed by leveraging both static and contextualized embeddings. In particular, s is determined by the linear combination of the scores obtained by two approaches: i) the APD measure over contextualized embeddings (see form-based approaches in Section 2.4.1); ii) the CD measure over static embeddings aligned according to the approach described by Hamilton et al. (2016). Similarly, in Martinc et al. (2020c), instead of directly using the APD measure, JSD is exploited over clusters of contextualized embeddings (see sense-based approaches in Section 2.4.2). As a further difference, the scores obtained by static and contextualized approaches are combined by multiplication. The intuition is that, since the score distributions of the two approaches are unknown, multiplication prevents an approach from contributing more than the other one in the final score.

Approaches can be also combined with grammatical profiles under the intuition that grammatical changes are slow and gradual, while lexical contexts can change very quickly (Kutuzov et al., 2021a; Giulianelli et al., 2022). Grammatical profile vectors gp_1 and gp_2 are associated with the times t_1 and t_2, respectively, to represent morphological and syntactical features of the considered language in the time period. Ryzhova et al. (2021) combine the contextualized embeddings of the word w occurrences with the grammatical vectors. A linear regression model with regularization is trained by using as features the cosine similarities over Φ_1 and Φ_2, and over the grammatical vectors gp_1 and gp_2.

As a further ensemble approach, the combination of different time-aware techniques such as temporal attention and time masking was tested by Rosin and Radinsky (2022) in order to better incorporate time into word embeddings.

2.4.4 Discussion

According to Section 2.4.1, 2.4.2, and 2.4.3, we note that form-based approaches are more popular than sense-based ones. Most papers are characterized by time-oblivious approaches and only a few time-aware approaches have recently appeared (e.g., Rosin and Radinsky, 2022). All approaches leverage unsupervised learning modalities with few exceptions (e.g., Hu et al., 2019). We argue that the motivation is due to the recent introduction of a reference evaluation framework for semantic change assessment proposed at SemEval-2020 Shared Task 1, where participants were asked to adopt an unsupervised configuration (Schlechtweg et al., 2020).

All papers are featured by contextualized word embeddings extracted from BERT-like models. Regardless of their version (i.e., tiny, small, base, large), BERT and XLM-R are the most frequently used LLMs, and only a few experiments rely on ELMo and RoBERTa. As a matter of fact, the size of data needed to train or fine-tune an XLM-R model is several orders of magnitude greater than BERT. Moreover, even if less frequently employed than BERT, ELMo seems to be promising for LSC and outperform BERT, while being much faster in training and inference (Kutuzov and Giulianelli, 2020). As a further interesting remark, the use of static *document* embeddings extracted from a Doc2Vec (Le and Mikolov, 2014) model has been proposed to provide pseudo-contextualized *word* embeddings as an alternative to BERT (Periti et al., 2022).

Monolingual and multilingual LLMs are both popular. The BERT models are the most frequently used monolingual models. XLM-R models are generally preferred to mBERT (i.e., multilingual BERT) models, since the former are trained on a larger amount of data and languages, thus the intuition is that they can better encode the language usages. Multilingual models are used both in multilingual settings, where corpora of different languages are considered (e.g., Martinc et al., 2020a), and monolingual settings, where just corpora of one language are given (e.g., Giulianelli et al., 2022). In a monolingual setting, the use of a multilingual model is motivated by two reasons: i) a model pre-trained on a specific language is not available (e.g., Kutuzov and Giulianelli, 2020), ii) multilingual models are employed to exploit their cross-lingual transferability property (e.g., Rachinskiy and Arefyev, 2021).

Considering the type of training, most of the papers directly use pre-trained LLMs or fine-tune them for domain adaptation. Only a few papers propose to exploit a specific fine-tuning (e.g., Pömsl and Lyapin, 2020) or to incrementally fine-tune a pre-trained LLM (e.g., Kutuzov and Giulianelli, 2020). Experiments indicate that fine-tuning a pre-trained LLM for domain adaptation consistently boosts the quality of results when compared against pre-trained LLMs (e.g., Qiu and Yang, 2022). The impact of fine-tuning on performance is analyzed by Martinc et al. (2020b), where it is shown that optimal results are achieved by fine-tuning a pre-trained LLM for five epochs and that, after five epochs, performance decreases due to over-fitting. However, we argue that the fine-tuning effectiveness strictly depends on the size and domain of the considered corpora. In many papers, a different number of epochs is proposed with varying results (e.g., Kutuzov and Giulianelli, 2020).

When a LLM is used, contextualized word embeddings are typically extracted from the last one or the last four layers of the model. Experiments show that the semantic features of text are mainly encoded in the last four encoder layers of BERT (Jawahar et al., 2019; Devlin et al., 2019). In some papers, contextualized embeddings are extracted by aggregating the output of the first and the last encoded layers. In this case, the idea is to combine *surface* features (i.e., phrase-level information, Jawahar et al., 2019) encoded in the first layer with the semantic features from the last one. Only Laicher et al. (2021) propose the standalone use of lower layers of BERT. Middle layers of BERT are usually excluded since they mainly encode syntactic features (Jawahar et al., 2019). When contextualized embeddings are extracted from more than one layer, they are generally aggregated by average or sum (e.g., Periti et al., 2022). As an alternative, the use of concatenation is proposed by Kanjirangat et al. (2020).

As a further note, when a LLM is used, some words may be split into word pieces by a subword-based tokenization algorithm (Wu et al., 2016; Sennrich et al., 2016). In this case, word piece representations are generally synthesized into a single word representation $e_{j,k}$ through averaging (e.g., Martinc et al., 2020a), or concatenating (e.g., Martinc et al., 2020c). As an alternative to avoid such a problem, the pre-trained vocabulary associated with the LLM can be extended by adding some words of interest. Then, a fine-tuning step is performed in order to learn the weights associated with the added words (e.g., Rosin et al., 2022).

Clustering operations are typically exploited in sense-based approaches to perform Word Sense Induction (Aksenova et al., 2022; Lau et al., 2012; Manandhar et al., 2010; Agirre and Soroa, 2007). The only form-based approach that relies on clustering is presented by Beck (2020) (see Section 2.4.1 for details). The clustering algorithms that are most frequently employed are K-Means and Affinity Propagation (AP). Further considered clustering algorithms are Gaussian Mixture Models (GMMs) (e.g., Rother et al., 2020), agglomerative clustering (AGG) (e.g., Arefyev and Zhikov, 2020), DBSCAN (e.g., Karnysheva and Schwarz, 2020), HDBSCAN (e.g., Rother et al., 2020), Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) (e.g., Rother et al., 2020), A-Posteriori affinity Propagation (APP) (e.g., Periti et al., 2022), and Incremental Affinity Propagation based on Nearest neighbor Assignment (IAPNA) (e.g., Periti et al., 2022). Since K-Means, GMMs, and AGG require to define the number of clusters in advance, the use of a silhouette score is generally employed to determine the optimal number of clusters (Rousseeuw, 1987). As an alternative, the AP algorithm is employed to let emerge the number of clusters without prefixing it. DBSCAN is proposed due to its capability of reducing noise by specifying i) the minimum number of embeddings of each cluster, and ii) the maximum distance ϵ between two embeddings in a cluster. HDBSCAN is the hierarchical version of DBSCAN and it can manage clusters of different sizes. As a difference with DBSCAN, HDBSCAN can detect noise without the ϵ parameter. APP and IAPNA are incremental extensions of AP, and their use is proposed for LSC when more than one time interval is considered. In Rother et al. (2020), different clustering algorithms are compared and the experiments show that i) DBSCAN is very sensitive to scale since ϵ is predefined, and ii) BIRCH tends to find a lot of small clusters that are marginal with respect to word meanings.

Considering the change functions, a detailed presentation of possible alternatives has been provided in Sections 2.4.1 and 2.4.2. As a final remark, we note that CD and APD are frequently exploited in form-based approaches, while JSD is commonly employed in sense-based approaches.

Finally, as for the language of considered corpora, most papers consider the shared benchmark datasets taken from competitive evaluation campaigns (e.g., LSCDiscovery, Zamora-Reina et al., 2022b). Commonly considered languages are English, German, Latin, and Swedish that appeared in 2020 at SemEval Task 1 (Schlechtweg et al., 2020). Russian appeared in 2021 at RuShiftEval (Kutuzov and Pivovarova, 2021b,c). Spanish appeared in 2022 at LSCDiscovery (Zamora-Reina et al., 2022b). The Italian language was introduced in 2020 at DIACRIta (Basile et al., 2020). The approach described by Martinc et al. (2020a) represents a novel attempt to consider a diachronic corpus containing texts of different languages, namely English and Slovenian.

2.5 Comparison of approaches on performance

In this section, we propose a comparison of the reviewed approaches based on their performance, considering the evaluation framework adopted in LSC tasks of shared competitions. The framework is based on a reference benchmark which contains a diachronic textual corpus in a given language. The framework is also characterized by a test-set of target words, where each word is associated with a continuous change score (i.e., *gold score*), typically calculated based on manual annotation following the established Word Usage Graph (WUG) paradigm (Schlechtweg et al., 2021).[1] Different metrics are also defined in the framework to evaluate the performance of the approaches according to the kind of assessment question that the task aims to address, namely *Grade/Binary Change*, *Sense Gain/Loss* (see Section 2.2).

In Table 2.7, we compare the reviewed approaches by considering the experiments on *Graded Change Detection* task performed and reported in the corresponding literature papers. In such a kind of task, the Spearman's correlation score is typically employed for assessing the performance of a given experiment by measuring the correlation between the predicted change scores and the gold scores.[2]. The Spearman's correlation evaluates the monotonic relationship between the rank order of the predicted scores and the gold ones. When multiple experiments are discussed in a paper, the best Spearman's correlation score obtained is reported in Table 2.7.

In the comparison, twelve diachronic corpora are exploited. In particular, we consider: i) the four SemEval datasets (Schlechtweg et al., 2020) for English (SemEval English), German (SemEval German), Latin (SemEval Latin), and Swedish (SemEval Swedish); ii) the English dataset proposed by Gulordava and Baroni (2011) (GEMS English); iii) the English LiverpoolFC dataset proposed by Del Tredici et al. (2019) (LivFC English); iv) the COHA English dataset (COHA English); v) the LSCDiscovery dataset (Zamora-Reina et al., 2022b) for Spanish (LSCD Spanish); vi) the DURel dataset for German (DURel German) (Schlechtweg et al., 2018); vii) the RuShiftEval dataset for Russian (RSE Russian) (Kutuzov and Pivovarova, 2021c); and viii) the NorDiaChange dataset for Norwegian (NOR Norwegian) (Kutuzov et al., 2022a). In Table 2.7, for each corpus, we highlight when a single time interval $C_1 - C_2$ or two consecutive time intervals $C_1 - C_2$ and $C_2 - C_3$ are considered, respectively. As a further remark, we note that the RSE Russian corpus is the only case where a test set for the time interval $C_1 - C_3$ as a whole is provided.

For the sake of readability, the performance according to the Spearman's correlation scores shown in Table 2.7 is labeled with the semantic change function of the considered approach and the corresponding framing with respect to form-based, sense-based, and ensemble-based categories (see Section 2.4).

As a general remark, we cannot find an approach outperforming all the others on all the considered cor-

[1] In the WUG annotation paradigm, human annotators provide semantic proximity judgments for pairs of word usages sampled from a diachronic corpus spanning two time periods. Word usages and judgments are represented as nodes and edges in a weighted, diachronic graph called *diachronic* WUG. This graph is then clustered with the correlation clustering algorithm (Bansal et al., 2004), and the resulting clusters are interpreted as *word senses*. Finally, for a given word, a ground truth score of semantic change is computed by comparing the probability distributions of clusters across different time periods, e.g., a cluster with most of its usages from one time period indicates a substantial semantic change.

[2] In Gonen et al. (2020), as an alternative to the Spearman's correlation score, the *Discount Cumulative Gain* is proposed. However, most papers still use Spearman's, since it is currently employed in competitive shared tasks.

	SemEval Englsh	**SemEval German**	**SemEval Latin**	**SemEval Swedish**	**GEMS English**	**LivFC English**	**COHA English**	**LSCD Spanish**	**DURel German**	**RSE Russian**			**NOR Norwegian**	
Ref.	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_2 - C_3$	$C_1 - C_3$	$C_1 - C_2$	$C_2 - C_3$
Teodorescu et al. 2022	-	-	-	-	-	-	-	*ensemble* APD .573	-	-	-	-	-	-
Zhou and Li 2020	*form* CD .392	*form* CD .392	*form* CD .392	*form* CD .392	-	-	-	-	-	-	-	-	-	-
Montariol et al. 2021	*sense* AP+WD .456	*sense* AP+JSD .583	*form* CD .496	*sense* K-Means+WD .332	***sense*** **AP+JSD** **.510**	-	-	-	*sense* AP+JSD .712	-	-	-	-	-
Periti et al. 2022	*sense* AP+JSD .514*	-	*sense* APP+JSD .512*	-	-	-	-	-	-	-	-	-	-	-
Pömsl and Lyapin 2020	*ensemble* APD .246	*ensemble* APD .725	*ensemble* APD .463	*ensemble* APD .546	-	-	-	-	***ensemble*** **APD** **.802**	-	-	-	-	-
Rachinskiy and Arefyev 2021	-	-	-	-	-	-	-	-	-	*ensemble* APD .781	*ensemble* APD .803	*ensemble* APD .822	-	-
Rachinskiy and Arefyev 2022	-	-	-	-	-	-	-	**sense** **APDP** **.745**	-	-	-	-	-	-
Rodina et al. 2021	-	-	-	-	-	-	-	-	-	*form* PRT .557	*sense* AP+JSD .406	-	-	-
Rosin et al. 2022	*form* CD .467	-	*form* CD .512	-	-	***form*** **TD** **.620**	-	-	-	-	-	-	-	-
Rosin and Radinsky 2022	***form*** **CD** **.627**	***form*** **CD** **.763**	*form* CD .565	-	-	-	-	-	-	-	-	-	-	-
Rother et al. 2020	*sense* HDBSCAN .512	*sense* GMMs .605	*sense* GMMs .321	*sense* HDBSCAN .308	-	-	-	-	-	-	-	-	-	-
Ryzhova et al. 2021	-	-	-	-	-	-	-	-	-	*ensemble* regression .480*	*ensemble* regression .487*	*ensemble* regression .560*	-	
Kudisov and Arefyev 2022	-	-	-	-	-	-	-	*form* APD .637	-	-	-	-	-	-
Kutuzov 2020	*form* APD .605	*form* PRT .740	*form* PRT .561	***form*** **APD** **.610**	*sense* AP+JSD .456*	-	-	-	-	-	-	-	-	-
Laicher et al. 2021	*form* APD .571*	*form* CD .755*	-	*form* APD .602*	-	-	-	-	-	-	-	-	-	-
Liu et al. 2021b	*form* CD .341	*form* CD .512	*form* CD .304	*form* CD .304	*form* CD .286	*form* CD .561	-	-	-	-	-	-	-	-
Martinc et al. 2020c	*ensemble* AP+JSD .361	*ensemble* AP+JSD .642	*form* CD .496	*ensemble* AP+JSD .343	-	-	-	-	-	-	-	-	-	-
Giulianelli et al. 2020	-	-	-	-	*form* APD .285*	-	-	-	-	-	-	-	-	-
Giulianelli et al. 2022	*form* APD .514	*ensemble* PRT .354	***ensemble*** **PRT** **.572**	*ensemble* APD .397	-	-	-	-	-	*ensemble* APD+PRT .376	*form* APD .480	*form* APD .457	***ensemble*** **APD+PRT** **.394**	***ensemble*** **APD** **.503**
Hu et al. 2019	-	-	-	-	-	-	***sense*** **MNS** **.428***	-	-	-	-	-	-	-
Kanjirangat et al. 2020	*sense* K-Means+JSD .028*	*sense* K-Means+JSD .173*	*sense* K-Means+JSD .253*	*sense* K-Means+CSC .321*	-	-	-	-	-	-	-	-	-	-
Karnysheva and Schwarz 2020	*sense* K-Means+JSD -.155*	*sense* DBSCAN+JSD .388*	*sense* DBSCAN+JSD .177*	*sense* K-Means+JSD -.062*	-	-	-	-	-	-	-	-	-	-
Kashleva et al. 2022	-	-	-	-	-	-	-	*sense* APDP .553	-	-	-	-	-	-
Keidar et al. 2022	*form* APD .489	-	-	-	-	-	-	-	-	-	-	-	-	-
Arefyev et al. 2021	-	-	-	-	-	-	-	-	-	***form*** **APD** **.825**	***form*** **APD** **.821**	***form*** **APD** **.823**	-	-
Arefyev and Zhikov 2020	*sense* AGG+CD .299	*sense* AGG+CD .094	*sense* AGG+CD -.134	*sense* AGG+CD .274	-	-	-	-	-	-	-	-	-	-
Beck 2020	*form* CD .293*	*form* CD .414*	*form* CD .343*	*form* CD .300*	-	-	-	-	-	-	-	-	-	-
Cuba Gyllensten et al. 2020	*form* CD .209*	*form* CD .656*	*form* CD .399*	*form* CD .234*	-	-	-	-	-	-	-	-	-	-
Kutuzov et al. 2022b	*form* APD .605	*form* PRT .740	*form* PRT .561	*form* APD .569	*form* APD .394	-	-	-	-	-	-	-	-	-

Table 2.7: The Spearman's correlation score of reviewed approaches in selected experiments. For each corpus, the top performance is reported in bold. Asterisks denote experiments based on a pre-trained model.

pora. This can suggest that an approach is language-dependent, namely it works well on one language and it is not appropriate for others. By relying on the experiments presented by Kutuzov and Giulianelli (2020), the performance of an approach is influenced by the employed assessment measure in relation to the distribution of the gold scores in the considered test set. The experiments by Kutuzov and Giulianelli (2020) show that when the distribution of the gold scores is skewed, namely some words are highly changed and some others are barely changed, the APD measure achieves better performance on Spearman's correlation than the PRT measure. On the contrary, when the distribution of the gold scores is almost uniform, namely most of the words are similarly changed, the PRT measure achieves better performance than the APD measure.

As a further remark, we note that the approaches characterized by fine-tuning achieve greater performance. This is also confirmed in the experiments of Martinc et al. (2020b) where fine-tuning a LLM boosts the performance when the LLM is not affected by under or over-fitting.

On average, form-based approaches outperform sense-based approaches in Graded Change Detection tasks. We argue that such a result is motivated by the structure of the test sets, where just one semantic change score is provided for each target word. Form-based approaches benefit from this structure since they work on measuring the change over one general word property (i.e., the dominant sense, or the degree of polysemy). On the opposite, sense-based approaches are disadvantaged by this structure since they work on measuring the change over multiple word meanings and they need to produce a single, comprehensive change value that summarises all the single-meaning changes for the comparison against the gold score. As a result, capturing some (minor) meanings can negatively affect the comprehensive change value, and to address this issue, small clusters are usually considered as possible noise and filtered out (Martinc et al., 2020c).

Table 2.7 shows that form-based approaches based on APD, CD, or PRT measures tend to obtain higher performance than sense- and ensemble-based approaches. GEMS English, COHA English, and LSCD Spanish are the only benchmarks where sense-based approaches outperform form-based ones. This can be motivated by the small number of experiments performed. Indeed, for COHA English experiments with form-based approaches have not been tested (Hu et al., 2019), while only a few experiments and a limited number of configurations with form-based approaches have been tested on GEMS English. For LSCD Spanish, the top performance is .745 and the corresponding approach leverages the APDP measure, which is an extension of APD characterized by the use of an average-of-average operation. This result is in line with the intuition presented by Periti et al. (2022), where the use of averaging on top of clustering contributes to reduce the noise in the contextualized embeddings of the target word.

We also note that ensemble approaches are on average characterized by high performance. In particular, top performances are provided by ensemble approaches on SemEval Latin (.572), DURel German (.802), and NOR Norwegian (.394 and .503). Notably, the performance on SemEval Latin is obtained by combining contextualized embeddings and grammatical profiles, thereby confirming that word meanings are influenced by morphology and syntax, especially in some languages. It is also interesting to observe that the performance on DURel German is obtained through an approach combining static and contextualized word embeddings, thus highlighting that such a kind of combination can be effective. For NOR Norwegian in the time interval $C_1 - C_2$, the best approach exploits both APD and PRT; this is a further confirmation that APD and PRT are top-performing measures in semantic change detection. For the subsequent time interval $C_2 - C_3$, the best result on NOR Norwegian is obtained with a combination of APD with grammatical profiles. This is a confirmation of the intuition presented by Giulianelli et al. (2022), which suggests that ensembling grammatical profiles with contextualized embeddings can enhance performance by incorporating morphological and syntactic features not fully captured by LLMs.

For SemEval English, SemEval German, the top performance are .627, .763, respectively, and they are obtained by the time-aware approach proposed by Rosin and Radinsky (2022). Also for LivFC English (.620), the top performance is obtained by leveraging a time-aware approach (Rosin et al., 2022). We argue that extra-linguistic information (e.g., time information) can have a positive impact on performance. The injection of extra-linguistic information can contribute to increase the performance also when small-size LLMs are employed, since they are less affected by noise than larger models. As a confirmation, in contrast to the widespread belief that the larger the models the higher the performance, the best result for SemEval English is obtained by exploiting contextualized embeddings generated from a BERT-tiny model (Turc et al., 2019; Rosin and Radinsky, 2022). This is also true for SemEval Swedish (.610), where the top performance is obtained by calculating the APD measure over contextualized embeddings extracted from an ELMo model (Kutuzov, 2020), which is far smaller than LLMs.

Finally, we note that also the use of supervised learning modalities contributes to achieve high performance. As an example, the top performances for RSE Russian are .825 on $C_1 - C_2$, .821 on $C_2 - C_3$, and .823 on $C_1 - C_3$ and they are obtained by a form-based, supervised approach (Arefyev et al., 2021). This is also confirmed by the recent introduction of a novel LLM called XL-LEXEME (Cassotti et al., 2023a), which has demonstrated exceptional performance across multiple benchmarks (Periti and Tahmasebi, 2024a).

2.6 Scalability, interpretability, and robustness issues

In this section, we analyze the LSC approaches by considering possible scalability, interpretability, and reliability issues.

2.6.1 Scalability issues

In the LSC approaches, any occurrence of the target word considered for change assessment is represented by a specific embedding. As a basic implementation, all the contextualized embeddings are stored in memory for processing. The higher the number of occurrences of a target word, the higher the number of embeddings to manage. As a result, when the size of the diachronic corpus grows, possible issues arise both in terms of memory and computation time. Similar issues occur when multiple target words are considered for change assessment. In this case, a possible workaround for addressing the memory issue is to process one target word at a time. However, in this way, the memory issue *changes* to a computation time issue. For feasibility convenience, most experiments work on a small set of target words. This kind of limitations inhibits the possibility to address tasks like the detection of the most changed word in a corpus. The need to work on solutions capable of dealing with such a kind of scalability issues has recently been promoted in LSCDiscovery, where participants were asked to assess the semantic change on all the words of the dictionary (Zamora-Reina et al., 2022b).

Some possible solutions to the scalability issues have been proposed in literature. For instance, approaches based on measures that enforce aggregation by averaging (e.g., CD, PRT) are time-scalable, since only the prototypes are considered for change assessment instead of the whole set of embeddings. Also approaches based on APD or JSD measures can be adjusted to become time-scalable. In particular, the number of embeddings to store and process can be reduced by random sampling the occurrences of the target word w. This means that i) a smaller number of similarity scores needs to be calculated with APD (e.g., Ryzhova et al., 2021), and ii) JSD works on top of clustering algorithms that converge faster (e.g., Rodina et al., 2021). As an alternative to random sampling, an online *aggregation by summing* method is proposed by Montariol et al. (2021), where a predefined number of contextualized embeddings n is stored in memory. An embedding e is stored when the number of embeddings in memory is less than n and e is strongly dissimilar from all the other embeddings previously stored. If e is not stored, it is aggregated to the most similar embedding stored in memory through sum.

The dimensionality reduction of the embeddings is proposed as a further alternative to enforce scalability. For example, in Rother et al. (2020), the embedding dimensionality is reduced to 10 (from 768) by combining an autoencoder with the UMAP (Uniform Manifold Approximation and Projection) algorithm (McInnes et al., 2020). In Keidar et al. (2022), UMAP and PCA are used to project contextualized embedding into $h \in \{2, 5, 10, 20, 50, 100\}$ dimensions. With respect to this solution, we argue that, although it can improve the memory scalability, time scalability is negatively affected since dimensionality reduction takes time. However, in (Rother et al., 2020), it is shown that the dimensionality reduction can still contribute to time scalability when the goal is to test and compare the effectiveness of different clustering algorithms and the reduced embeddings are saved and re-used. As a further option, the use of small LLMs, such as TinyBert or ELMo, is gaining more and more attention since the dimension of the generated embeddings is far lower (e.g., Rosin and Radinsky, 2022).

Scalability issues can also arise when the change needs to be assessed on a corpus $C = \bigcup_i^n C_i$ defined over more than one time interval ($n > 2$). Typically, existing approaches calculate the change score s over each pair of time intervals (t_i, t_{i+1}) by iteratively re-applying the same assessment workflow. As a difference, an incremental approach based on a clustering algorithm called *A Posteriori affinity Propagation* (APP) is proposed by Castano et al. (2024) and Periti et al.; Periti et al. (2024e; 2022) to speed up the aggregation stage. In each time interval, clustering is incrementally executed by considering the prototypes of the previous time period (i.e., aggregation by averaging) and the incoming embeddings of the current time period.

2.6.2 Interpretability issues

Interpretability issues arise when it is not possible to determine which meaning(s) have changed among all the meanings of a target word, namely the meaning(s) that mainly caused the change score assessed by a considered approach. Definitely, form-based approaches are affected by such a kind of issues, since they model the change as the change in the dominant sense or in the degree of polysemy of a word, without considering the possible multiple meanings. On the opposite, sense-based approaches aim at providing an interpretation of the word change, since they attempt to model the change by considering the multiple word senses. However, interpretability issues can arise also when sense-based approaches are employed due to three main motivations.

Word meaning representation. Sense-based approaches mostly rely on clustering techniques to represent word meanings. The K-Means and the AP clustering algorithms are usually employed to this end. K-Means requires that the number of target clusters is predefined, and this can be inappropriate to effectively represent the meanings of a target word that are not known beforehand. AP lets the number of target clusters emerge, but experimental results show that the association of a cluster with a word meaning can be imprecise. We argue that this can be due to the distributional nature of LLMs that tends to capture changes in contextual variance (i.e., word usages) rather than changes in lexicographic senses (i.e., word meanings) (Kutuzov et al., 2022b). As an example, sometimes AP produces more than 100 clusters, which is rather unrealistic if we assume that a cluster represents a word meaning (Periti et al., 2022). As a matter of fact, a word may completely change its context without changing its meaning (Martinc et al., 2020b).

Word meaning description. Each cluster obtained during the aggregation stage of a sense-based approach needs to be associated with a description that denotes the corresponding word meaning. This can be done by human experts on the basis of the cluster contents. However, this is time-consuming, given that a cluster can consist of several hundreds/thousands of elements. As an alternative, clustering analysis techniques have been proposed to label clusters by summarizing their contents. As a possible option, a cluster description can be extracted from the content by considering the top featuring keywords based on lexical occurrences (e.g., Tf-Idf) (Kellert and Mahmud Uz Zaman, 2022; Montariol et al., 2021) or substitutes (Card, 2023). In (Giulianelli et al., 2020), the sense-prototype of a cluster is proposed as a cluster exemplar and the corpus sentences that are closest to the prototype are adopted as cluster/meaning description. However, when a cluster contains outliers, these sentences could not provide an effective description. More recently, the use of Causal LLMs has been proposed to generate descriptive cluster interpretations (Castano et al., 2024) or word usage definitions (Giulianelli et al., 2023).

Word meaning evolution. When a corpus $C = \bigcup_i^n$ defined over more than one time interval is considered, the clusters defined at a time step t_i need to be linked to the clusters of the previous time step t_{i-1} to trace the evolution of the corresponding meaning over time (i.e., cluster/meaning history). Since the clustering executions at each time step are independent, the capability of recognizing corresponding clusters/meanings at different time steps can be challenging. As a possible solution, alignment techniques can be employed to link similar word meanings in different, consecutive time periods (Kanjirangat et al., 2020; Montariol et al., 2021). As a further option, evolutionary clustering algorithms can be exploited without requiring any alignment mechanism across time periods (Castano et al., 2024; Periti et al., 2024e, 2022).

2.6.3 Robustness issues

Robustness issues arise when the assessment score is not reliable due to data imbalance, model stability, and model bias.

Data imbalance. The diachronic corpus C must equally reflect the presence of the target word w in both the time steps t_1 and t_2. This means that the frequency of w must not strongly change in the considered time period. However, in common scenarios, more documents are available for the most recent time step t_2 and *"it may not be possible to achieve balance in the sense expected from a modern corpus"* (Tahmasebi et al., 2021a). As a consequence, the frequency of w can be strongly higher in t_2 than in t_1 and the embeddings Φ_j can produce a distorted representation of the target word when the LLM is trained/fine-tuned (e.g., Wendlandt et al., 2018; Zhou et al., 2021). As a further remark, data imbalance issues can occur when some word meanings are more frequent than others. For instance, the dominant sense is usually more represented than other senses in the corpus C. As a result, when a sense-based approach is adopted, the embedding distributions p_1, p_2 can be skewed, meaning that a larger number of embeddings is associated with the dominant sense rather than with the other minor senses. In sense-based approaches, the word meanings are represented by clusters, and *the number of clusters consistently reflects word frequency* (Kutuzov, 2020). When a meaning is associated with a few embeddings/clusters, its contribution to the overall assessment score is marginally leading to an inflated or underestimated assessment score. In this respect, a qualitative analysis of "potentially erroneous" outputs of reviewed approaches is presented by Kutuzov et al. (2022b). Some examples of potentially erroneous assessment scores occur when i) a *word with strongly context-dependent meanings* is considered, whose embeddings are mutually different; ii) a *word is frequently used in a very specific context* in only one time step t_1 or t_2; iii) a *word is affected by a syntactic change*, not a semantic one. Liu et al. (2021b) propose a solution to reduce the false discovery rate and to improve the precision of the change assessment by leveraging permutation-based statistical tests and term-frequency thresholding.

Model stability. Pre-trained LLMs are usually trained on modern text sources. For example, the original English BERT model is pre-trained on Wikipedia and BooksCorpus (Zhu et al., 2015). As a result, pre-trained LLMs are prone to represent words from a modern perspective, and thus they tend to ignore the temporal information of a considered corpus. This way, when historical corpora are considered, the possible obsolete word usages cannot be properly represented. This problem has been investigated in the literature by comparing the performance of pre-trained against fine-tuned LLMs (Kutuzov and Giulianelli, 2020; Qiu and Yang, 2022). In line with the considerations of Section 2.4.4, the results show that fine-tuning the LLM on the whole diachronic corpus improves the quality of word representations for historical texts. Since fine-tuning the LLM can be expensive in terms of time and computational resources, a measure for estimating the model effectiveness for historical sources is presented by Ishihara et al. (2022). In particular, this measure is used to decide whether a model should be re-trained or fine-tuned.

Model bias. Contextualized embeddings can possibly be affected by biases on the encoded information. For instance, a possible bias can arise from orthographic information, such as the word form and the position of a word in a sentence, since they influence the output of the top BERT layers (Laicher et al., 2021). Text pre-processing techniques are proposed as a solution to reduce the influence of orthography in the embeddings, thus increasing the robustness of encoded semantic information. To this end, lower-casing the corpus text is a commonly employed solution. However, *the lower-casing of words often conflates parts of speech*, thus another possible bias can arise. For example, the proper noun `Apple` and the common noun `apple` become identical after lower-casing (Hengchen et al., 2021). The possible bias introduced by Named Entities and proper nouns is investigated by Laicher et al.; Martinc et al. (2021; 2020c). In Qiu and Yang (2022), text normalization techniques are proposed based on the removal of accent markers. In some languages, such a kind of normalization can introduce a bias since different words can be conflated. For example, `papà` (e.g., the Italian word for `dad`) and `papa` (e.g., the Italian word for `pope`) cannot be distinguished after the accent removal. Further text pre-processing techniques can be employed to reduce the possible bias due to orthographic information. In Schlechtweg et al. (2020), lemmatization and punctuation removal are proposed. Experimental results on lemmatization for reducing the model bias on BERT embeddings are presented by Laicher et al. (2021). Further experiments show that lemmatizing the target word alone is more beneficial than lemmatizing the whole corpus (Laicher et al., 2021). Filtering out content-light words, such as stop words and low-frequency words, can be also beneficial (Zhou and Li, 2020). As an alternative solution to reduce word-form biases, the embedding of a word occurrence can be computed by averaging its original embedding and the embeddings of its nearest words in the input sentence (Zhou and Li, 2020).

When aggregation by clustering is enforced, the possible word-form biases can affect the clustering result (Laicher et al., 2021). As a solution, clustering refinement techniques have been proposed. As an option, the removal of the clusters containing only one or two instances is adopted, since they are not considered significant (Martinc et al., 2020c). As a further option, in Martinc et al. (2020b), clusters with less than two members are considered as weak clusters and they are merged with the closest strong cluster, i.e. cluster with more than two members. In Periti et al. (2022), clusters containing less than 5 percent of the whole set of embeddings are assumed to be poorly informative and are thus dropped. However, we argue that the use of clustering refinement techniques must be carefully considered since also small clusters can be important when the corpus is unbalanced in the number of meanings of a word.

2.7 Challenges and considerations

In this chapter, we analyzed the LSC task by providing a formal definition of the problem, and a reference classification framework based on meaning representation, time awareness, and learning modality dimensions. The literature approaches are surveyed according to the given framework by considering the assessment function, the employed LLM, the achieved performance, and the possible scalability/interpretability/robustness issues.

While we provide a solid framework for classification LSC approaches, we acknowledge that the NLP research on semantic change is rapidly evolving with new papers continually emerging. For example, various models such as LLaMA (Periti et al., 2024b), GPT (Periti and Tahmasebi, 2024a), and ChatGPT (Periti et al., 2024d) are being considered for LSC. Approaches based on lexical substitutes are gaining popularity to analyze both the modern and the historical bias of LLMs (Cuscito et al., 2024). Further supervised (Tang et al., 2023) and unsupervised (Aida and Bollegala, 2023) approaches, along with different change functions (Aida and Bollegala, 2024) are appearing. Additionally, new benchmarks for a larger gamma of languages are becoming available, including Chinese (Chen et al., 2022a, 2023a), Japanese (Ling et al., 2023), and Slovenian (Pranjić et al., 2024).

In Hengchen et al.; Kutuzov et al. (2021; 2018), an overview of open challenges for LSC is presented. In the following, we extend such an overview by focusing on those challenges that are specific to the existing approaches in relation to the issues discussed in Section 2.6.

Scalability. The trend in LSC is to adopt increasingly larger models with the idea that they better represent language features. As a consequence, scalability issues arise, and they are being addressed as discussed in Section 2.6.1. However, contrary to this trend, we argue that the use of small-size models, such as those introduced by Rosin and Radinsky; Rosin et al. (2022; 2022), needs to be further explored since they are competitive in terms of performance.

Word meaning representation. In Section 2.5, we show that form-based approaches outperform sense-based approaches in the Graded Change Detection assessment. However, we argue that sense-based approaches are promising since they focus on encoding word senses and they can enrich the mere degree of semantic change of a word w with the information about the specific meaning of w that changed. In this direction, LSC should be considered as a temporal/diachronic extension of other problems such as Word Sense Induction (Alsulaimani et al., 2020), Word Meaning Disambiguation (Godbole et al., 2022), and Word-in-Context (Loureiro et al., 2022).

In this regard, we will connect LSC to other problems in Chapter 3 (LSC through Word-in-Context), Chapter 4 (LSC through Word Sense Induction), and more formally in Chapter 7 and Chapter 9 (LSC as Word-in-Context + Word Sense Induction + Graded Change Detection).

So far, word senses have been represented through aggregation by clustering under the idea that each cluster represents a specific word meaning. However, according to the interpretability issues of Section 2.6, clustering techniques are often affected by noise and they are typically capable of representing word usages rather than word meanings. Thus, further investigations are required to represent lexicographic meanings in a more faithful way.

Word meaning description. According to Section 2.6, current solutions to meaning description are focused on determining a representative label taken from the cluster contents (e.g., Tf-Idf, sentence(s) featuring the sense-prototype). Such solutions are mostly oriented to highlight the lexical features of the cluster/meaning without considering any element that reflects the cluster's semantics. As a consequence, open challenges are based on the need of comprehensive description techniques capable of capturing both lexical and semantic aspects such as position in text, semantics, or co-occurrences across different documents. In a very recent work, (Giulianelli et al., 2023) propose interpreting the meaning of word usages by generating sense definitions through novel generative models. A main drawback is that different definitions can be generated for usages related to the same meaning. Nonetheless, we strongly suggest a change towards the latter solution, given that the new generative models have demonstrated extraordinary capabilities.

In this regard, we will preliminarily investigate the use of generative LLMs in Chapter 3 and Chapter 7, and more extensively in Chapter 8 and Chapter 9.

Word meaning evolution. In shared competitions, the reference evaluation framework for LSC is based on one/two time periods that are considered for LSC. The extension of the evaluation framework to consider more time periods is an open challenge. In particular, methods and practices of LSC approaches need to be tested/extended for detecting both short- and long-term semantic changes as well as for promoting the design of incremental techniques able to handle dynamic corpora (i.e., corpora that become progressively available).

In this context, a further challenge is about the capability to trace the change of a meaning over multiple time steps (i.e., meaning evolution). As mentioned in Section 2.2, alignment techniques can be used to link similar word meanings in different, consecutive time periods. However, such a solution is not completely satisfactory due to possible limitations (e.g., scalability, robustness of alignment), and further research work is needed to better track the meaning evolution over time (e.g., Periti et al., 2022).

In this regard, we will further discuss this challenge in Chapter 4 and present a novel incremental approach to LSC in Chapter 5.

Model stability. Most of the approaches surveyed in this chapter are time-oblivious and face the problem of model stability through fine-tuning. Since this practice can be expensive in terms of time and resources, we argue that further research on the development of time-aware approaches is needed, in that, they do not suffer the model stability problem.

In this regard, in Chapter 8 we will leverage lexical replacements to evaluate the contextualization capability of LLMs when lexical semantic change occurs.

Model bias. The solutions to model bias issues presented in Section 2.6 are language-dependent and they are mainly exploited in approaches based on monolingual models. Further research work is needed to test the effectiveness of existing solutions also in approaches based on multilingual LLMs. In addition, we argue that future work should concern the application of denoising and debiasing techniques to both monolingual and multilingual LLMs (e.g., Kaneko and Bollegala, 2021) with the aim to improve LSC performance by reducing orthographic biases regardless of the language(s) on which the models were trained.

Further challenges not strictly related to the issues of Section 2.6 are the following:

Semantic Change Interpretation. Most of the literature does not investigate the nature of the detected change, meaning that they do not classify the semantic change according to the existing linguistics theory (e.g., amelioration, pejoration, broadening, narrowing, metaphorization, metonymization, and metonymy) (Campbell, 2020; Hock and Joseph, 2019). Further studies on the causes and types of semantic changes are needed (de Sá et al., 2024). These studies could be crucial to detect "laws" of semantic change that describe the condition under which the meanings of words are prone to change. For example, some laws are hypothesized or tested by Xu and Kemp (2015); Dubossarsky et al. (2015); Hamilton et al. (2016), but later the validity of some of them has been questioned (Dubossarsky et al., 2017). Contextualized embeddings could contribute to test the validity of current laws and to propose new ones. To the best of our knowledge, some steps in this direction are only moved by (Hu et al., 2019) for modeling the word change from an ecological viewpoint (similar to the dynamics of species populations over time).

Computational models of meaning change. Almost all experiments on LSC are based on BERT embeddings. Although there are open questions about how to maximize the effectiveness of BERT embeddings in different language setups, the effectiveness of BERT for LSC has been extensively investigated. We believe that LSC should be extended by considering a wider range of models. Some work explored the effectiveness of ELMo (Kutuzov and Giulianelli, 2020; Rodina et al., 2021). However, the performance of ELMo in different contexts and setups should be analyzed in more detail. Furthermore, it might be worth investigating smaller versions of BERT, like ALBERT (Lan et al., 2019) and DistilBERT (Sanh et al., 2019). Further models can also be considered like seq2seq and generative models, which recently showed interesting results in the field of temporal Word-in-Context problem (Lyu et al., 2022).

In this regard, we will evaluate the use of GPT-3.5 in Chapter 3, compare the use of BERT, mBERT, XLM-R, XL-LEXEME, and GPT-4 in the systematic evaluation presented in Chapter 7, and investigate the use of LLaMA in Chapter 8- 9 and Flan-T5 in Chapter 9.

Multilingual models. In LSC shared competitions, monolingual models have generally been preferred to multilingual ones. We believe that a systematic comparison of monolingual vs. multilingual models is required to determine scenarios and conditions where the former type of models provides better performance than the latter type or vice-versa. Multilingual embeddings can also contribute to LSC since they could enable a language-independent semantic change assessment, meaning that the gold scores of different languages can be exploited as a whole for the evaluation of a given approach.

In this regard, we will thoroughly compare the use of monolingual models against multilingual models for LSC in the systematic evaluation presented in Chapter 7.

Cross-language change detection. As introduced by Martinc et al. (2020a), further investigations are required to address the problem of cross-language change detection. We argue that solutions to such a kind of problem can be also useful for LSC since they can detect semantic change of *cognates* and *borrowings* (e.g., Fourrier and Montariol, 2022), as well as *contact-induced* semantic changes (e.g., Miletic et al., 2021)[3].

Use cases. So far, LSC through contextualized embeddings is still a theoretical problem not yet integrated into real application scenarios like historical information retrieval, lexicography, linguistic research, or social analysis. Among the existing use cases, semantic change has been examined by Bonafilia et al. (2023) to investigate sudden events that radically alter public opinion on a topic, and by Menini et al.; Paccosi et al. (2022; 2023) to explore shifts in olfactory perception and changes in the descriptions of smells over time. We expect that further use cases and experiences will developed and shared in the future.

Context change over different domains. The attention gained by diachronic semantic change detection through the use of word embeddings paved the way for modeling other linguistics issues such as the identification of diatopic lexical variation (Seifart, 2019), the detection of semantic changes of grammatical constructions (Fonteyn et al., 2020), or the comparison of how speakers who disagree on a subject use the same words (Garí Soler et al., 2022). The reviewed approaches can be tested and possibly extended to cope with such a kind of linguistics issues.

[3] In linguistics, cognates are sets of words in different languages that have been inherited in direct descent from an etymological ancestor in a common parent language. Borrowings (or loanwords) are words adopted by the speakers of one language from a different language. Contact-induced semantic changes are diachronic changes within a recipient language that are traceable to languages other than the direct ancestor of the recipient language and that have spread and are conventionalized within a community speaking the recipient language.

Chapter 3

A very first evaluation of ChatGPT

> "*The Answer to the Great Question... Of Life, the Universe and Everything... Is... Forty-two*"
>
> Douglas Adams, *The Hitchhiker's Guide to the Galaxy*

3.1 Introduction

The recent introduction of Transformer-based (Vaswani et al., 2017) language models has led to significant advances in NLP. These advances are exemplified in Pre-trained Foundation LLMs like BERT and GPT, which "*are regarded as the foundation for various downstream tasks*" (Zhou et al., 2023a).

In the previous chapter, we reviewed the current state-of-the-art for LSC, presenting approaches mainly based on encoder-based LLMs. Among them, BERT has experienced a surge in popularity over the last few years, and the family of BERT models has repeatedly provided state-of-the-art (SOTA) results for LSC. However, with the introduction of ChatGPT, research attention began shifting towards generative models, particularly ChatGPT due to its impressive ability to generate fluent and high-quality responses to human queries. Within just five days of its release on November 30, 2022, ChatGPT attracted 1 million users. This rapid adoption continued, surpassing 100 million users by January 2023, making it the fastest-growing application in history. As of 2024, its user base has now exceeded 180.5 million.

Several recent research studies have assessed the language capabilities of ChatGPT by using a wide range of prompts to solve popular NLP tasks (Laskar et al., 2023; Kocoń et al., 2023). However, current evaluations generally (a) overlook that the output of ChatGPT is nondeterministic,[1] (b) rely only on contemporary and *synchronic* text, and (c) consider predictions generated by the ChatGPT[2] web interface, whose parameter settings were initially unknown at the time of this thesis. As a result, these evaluations provide valuable insights into the generative, pragmatic, and semantic capabilities of ChatGPT (Kocoń et al., 2023), but fall

[1] platform.openai.com/docs/guides/gpt/faq

[2] chat.openai.com

short when it comes to assess the potential of ChatGPT to solve NLP tasks and specifically to handle *historical* and *diachronic* text, which constitutes a unique scenario for testing models' capability to generalize.

Chapter outline.

This chapter includes materials originally published in the following publication:

> Francesco Periti, Haim Dubossarsky, and Nina Tahmasebi. 2024**d**. (Chat)GPT v BERT: Dawn of Justice for Semantic Change Detection. In Findings of the Association for Computational Linguistics: EACL 2024, pages 420–436, St. Julian's, Malta. Association for Computational Linguistics.

In this chapter, we propose to evaluate the use of both ChatGPT Web and ChatGPT API[3] - to recognize lexical semantic change. Our goal is not to comprehensively evaluate ChatGPT in dealing with semantic change but rather to evaluate its potential as *off-the-shelf* model with a *reasonable* prompts from a human point of view, which may not necessarily be optimized for the model. The chapter is organized as follows. Section 3.2 frames our evaluation within the relevant literature of its time. Section 3.3 outlines our evaluation setup and introduces the considered evaluation questions. The results of our evaluation are presented in Section 3.4. Finally, we discuss our experimental evaluation in Section 3.5.

3.2 Background and related work

As this thesis progresses, a continuous stream of research has been published in parallel and continues to emerge, given that ChatGPT has become a hot topic. In light of this, we provide a concise overview that reflects the current landscape at the time of our evaluation study. Our intention is not to present an exhaustive review, but rather to highlight central concerns observed in prior evaluations.

3.2.1 Related work

The significant attention garnered by ChatGPT has led to a large number of studies being published immediately after its release. Early studies mainly focused on exploring the benefits and risks associated with using ChatGPT in expert fields such as education (Lund and Wang, 2023), medicine (Antaki et al., 2023), or business (George and George, 2023). Evaluation studies are currently emerging for assessing (Chat)GPT's generative and linguistic capabilities across a wide range of downstream tasks in both monolingual and multilingual setups (Bang et al., 2023; Shen et al., 2023; Lai et al., 2023). Most evaluations focus on ChatGPT and involve a limited number of instances (e.g., 50) for each task considered (Weissweiler et al., 2023; Zhong et al., 2023; Alberts et al., 2023; Khalil and Er, 2023). When the official API is used to query ChatGPT, this limit is imposed by the hourly token processing limit[4] and the associated costs.[5] When the web interface

[3] Throughout the text, we represent instances of both ChatGPT Web and API as ChatGPT.

[4] help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them

[5] openai.com/pricing

is used instead of the API, the limit is due to the time-consuming process of interacting with ChatGPT that keeps humans “in the loop”. Thus far, even systematic and comprehensive evaluations (Kocoń et al., 2023; Laskar et al., 2023) rely on the repetition of a single experiment for each task. However, while individual experiments provide valuable insights into ChatGPT's capabilities, they fall short in assessing the potential of capabilities to solve specific tasks given its nondeterministic nature. Multiple experiments need to be conducted to validate its performance on each task. In addition, current evaluations generally leverage tasks that overlook the temporal dimension of text, leaving a gap in our understanding of ChatGPT's capability to handle diachronic and historical text.

3.2.2 Evaluating ChatGPT through Word-in-Context

The LSC modeling presented in Chapter 2 involves considering all occurrences (potentially several thousand) of a set of target words to assess their change in meaning within a diachronic corpus. As a result, this setup is *currently* not suitable for evaluating ChatGPT, due to the limited size of its prompts and answers, as well as accessibility limitations such as an hourly character limit and economic constraints. In light of these considerations, we chose to evaluate the potential of ChatGPT through the Word-in-Context (WiC) task, which has recently demonstrated a robust connection with LSC (Cassotti et al., 2023a; Arefyev et al., 2021). Considering the remarkable performance of contextualized BERT models in addressing WiC and LSC tasks (Periti and Montanelli, 2024; Periti and Dubossarsky, 2023; Periti et al., 2024e), we compare the performance of ChatGPT to those obtained using BERT.

Our evaluation of ChatGPT focuses on a diachronic extension of the original WiC setting. In particular, we evaluate ChatGPT to determine whether a word carries the same meaning in two different contexts of different time periods, or conversely, whether those contexts exemplify a semantic change. Our aim is to assess the potential of ChatGPT for LSC, offering the first investigation into the application of ChatGPT for historical linguistic purposes. Prior to our evaluation, only the use of ChatGPT for a conventional WiC task has been evaluated by Laskar et al. (2023) and Kocoń et al. (2023), who reported low accuracy under a single setup. Our evaluation challenges their performance by considering diachronic text and different setups.

3.3 Evaluation setup

In the following, we first present the WiC problem and the diachronic benchmarks used for our evaluation. Then, we outline our evaluation questions (EQs) along with the various setups we considered.

Problem statement. The original WiC task is framed as a binary classification problem, where each instance is associated with a target word w, either a verb or a noun, for which two contexts, c_1 and c_2, are provided (Pilehvar and Camacho-Collados, 2019). The task is to identify whether the occurrences of w in c_1 and c_2 correspond to the same meaning or not. Both TempoWiC and HistoWiC rely on the same definition of the task, while being specifically designed for semantic change detection in diachronic text.

Benchmarks. In our evaluation, we consider two diachronic WiC benchmarks, namely *temporal* WiC (TempoWiC, Loureiro et al., 2022) and *historical* WiC (HistoWiC). While TempoWiC has been designed to evaluate LLMs ability to detect short-term change in social media, HistoWiC is our adaptation of the SemEval benchmark of historical text to a WiC task for evaluating LLMs ability to detect long-term change in historical corpora.

	TempoWiC			**HistoWiC**		
	Trial	Train	Test	Trial	Train	Test
True	8	86	73	11	137	79
False	12	114	127	*9*	103	61
Total	20	200	200	20	200	140

Table 3.1: Datasets used in our evaluation.

- **Temporal Word-in-Context.** NLP models struggle to cope with new content and trends. TempoWiC is designed as an evaluation benchmark to detect short-term semantic change on social media, where the language is extremely dynamic. It uses tweets from different time periods as contexts c_1 and c_2.

  Given the limits on testing ChatGPT, we followed Zhong et al. (2023); Jiao et al. (2023) and randomly sampled a subset of the original TempoWiC datasets. While the original TempoWiC framework provides Trial, Train, Test, and Dev sets, here we did not consider the Dev set. Table 3.1 shows the number of positive (i.e., same meaning) and negative (i.e., different meanings due to semantic change) examples we considered for each set.

- **Historical Word-in-Context.** Given that NLP models also struggle to cope with historical content and trends, we designed HistoWiC as a novel evaluation benchmark for detecting long-term semantic change in historical text, where language may vary across different epochs. HistoWiC sets the two contexts, c_1 and c_2, as sentences collected from the two English corpora of the LSC detection task (Schlechtweg et al., 2020).

  Similar to the original WiC (Pilehvar and Camacho-Collados, 2019), the annotation process for the SemEval-English benchmark involved usage pair annotations where a target word is used in two different contexts. Thus, we directly used the annotated instances of LSC to develop HistoWiC. Since LSC instances were annotated using the DURel framework (Schlechtweg et al., 2024) and a four-point semantic-relatedness scale (see Table 3.2), we only binarized the human annotations. As with TempoWiC, we randomly sampled a limited number of instances to create Trial, Train, and Test sets. Table 3.1 shows the number of positive and negative examples for each set.

In particular, for HistoWiC, we shifted from the LSC to the WiC setting as follows. First, we selected only the annotated LSC instances containing contexts from different time periods. We then filtered out all the instances annotated by a single annotator[6] and all the instances that are associated with an average score, s, such that $1.5 < s < 3.5$, which represents ambiguous cases even for humans. Finally, we binarized the LSC annotations by converting each $s \leq 1.5$ to *False* (i.e. different meanings) and each $s \geq 3.5$ to *True* (i.e. same meaning). We report in Table 3.2 the scale used to annotate the LSC instances through the DURel framework.

As an example, consider the following word usage pair $\langle w, c_1, c_2 \rangle$ extracted by the SemEval-English benchmark for the word w =*plane*.

c_1: But we are most familiar with the exhibitions of gravity in bodies descending inclined **planes**, as in the avalanche and the cataract.

c_2: Over the next several years, he said, the Coast Guard will get 60 more people, two new 270-foot vessels and al twin-engine **planes**.

Following the DURel scale, the pair has been annotated with an average judgment of 1 by human annotators. We thus converted this judgment to *False*.

↑	4:	Identical
	3:	Closely related
	2:	Distantly related
	1:	Unrelated

Table 3.2: The DURel relatedness scale used in Schlechtweg et al.; Schlechtweg; Schlechtweg et al.; Schlechtweg et al.; Schlechtweg et al.; Schlechtweg and Schulte im Walde; Schlechtweg et al. (2024; 2023; 2021; 2020; 2018; 2020; 2018).

3.3.1 Evaluation questions

In our experiments, we evaluated the performance of ChatGPT-3.5 over the TempoWiC and HistoWiC Test sets using both the official OpenAI API (API)[7] and the web interface (Web).[8] Of the GPT-3.5 models available through the API, we assessed the performance of gpt-3.5-turbo. Following Loureiro et al. (2022), we employed the Macro-F1 for multi-class classification problems as evaluation metric.

Different prompts. Current ChatGPT evaluations are typically performed manually (Laskar et al., 2023). When automatic evaluations are performed, they are typically followed by a manual post-processing procedure (Kocoń et al., 2023). As manual evaluation and processing may be biased due to answer interpretation, we addressed the following evaluation question:

EQ1: *Can we evaluate ChatGPT in WiC tasks in a completely automatic way?*

[6] Different instances were annotated by varying numbers of annotators.

[7] version 0.27.8.

[8] The August 3 Version.

Furthermore, as current evaluations generally rely on a zero-shot prompting strategy, we addressed the following evaluation question:

> **EQ2:** *Can we enhance ChatGPT's performance in WiC tasks by leveraging its in-context learning capabilities?*

To address EQ1 and EQ2, we designed a prompt template (see Table 3.3) to explicitly instruct ChatGPT to answer in accordance with the WiC label format (i.e., *True*, *False*). We then used this template (see Table 3.4) with different prompt strategies:

- *zero-shot prompting* (ZSp): ChatGPT was asked to address the WiC tasks (i.e., Test sets) without any specific training, generating coherent responses based solely on its pre-trained knowledge.
- *few-shot prompting* (FSp): since LLMs have recently demonstrated *in-context learning* capabilities without requiring any fine-tuning on task-specific data (Brown et al., 2020), ChatGPT was presented with a limited number of input-output examples (i.e., Trial sets) demonstrating how to perform the task. The goal was to leverage the provided examples to improve the model's task-specific performance.
- *many-shot prompting* (MSp): similar to FSp, but with a greater number of input-output examples (i.e., Train sets).

Description	Template
task explanation	**Task**: Determine whether two given sentences use a target word with the same meaning or different meanings in their respective contexts.
explicit behavioral guidelines	I'll provide some negative and positive examples to teach you how to deal with the task before testing you. Please respond with only "OK" during the examples; when it's your turn, answer only with "True" or "False" without any additional text. When it's your turn, choose one: "True" if the target word has the same meaning in both sentences; "False" if the target word has different meanings in the sentences. I'll notify you when it's your turn.
example instance	This is an example. You have to answer "OK": **Sentence 1**: [First sentence containing the target word] **Sentence 1**: [First sentence containing the target word] **Target**: [Target word] **Question**: Do the target word in both sentences have the same meaning in their respective contexts? **Answer**: [True/False]
task instance	Now it's your turn. You have to answer with "True" or "False": **Sentence 1**: [First sentence containing the target word] **Sentence 1**: [First sentence containing the target word] **Target**: [Target word] **Question**: Do the target word in both sentences have the same meaning in their respective contexts? **Answer**: [The model is expected to respond with "True" or "False"]

Table 3.3: Sections of the prompt template used for testing (Chat)GPT.

Varying temperature. The temperature is a hyper-parameter of ChatGPT that regulates the variability of responses to human queries. According to the OpenAI FAQ, the temperature parameter ranges from 0.0

ID	Strategy	Prompt
ZSp	zero-shot prompting	task explanation explicit behavioral guidelines task instance ... task instance
FSp	few-shot prompting	task explanation explicit behavioral guidelines example instance ... example instance task instance ... task instance
MSp	many-shot prompting	*like FSp*

Table 3.4: Prompt template for each employed prompting strategy.

to 2.0, with lower values making outputs mostly deterministic and higher values making them more random.[9] To counteract the non-determinism of ChatGPT, we focused only on TempoWiC and HistoWiC and conducted the same experiment multiple times with progressively increasing temperatures. This approach enabled us to answer the following evaluation questions:

EQ3: *Does ChatGPT demonstrate comparable effectiveness in detecting short-term change in contemporary text and long-term change in historical text?*

EQ4: *Can we enhance ChatGPT's performance in WiC tasks by raising the "creativity" using the temperature value?*

To address EQ3 and EQ4, we evaluated ChatGPT API in TempoWiC and HistoWiC using eleven temperatures in the range [0.0, 2.0] with 0.2 increments. For each temperature and prompting strategy, we performed two experiments and considered the average performance.

Comparing ChatGPT API and Web. Current evaluations typically prompt ChatGPT through the web interface instead of the official OpenAI API. This preference exists because the web interface is free and predates the official API. However, there are differences between using ChatGPT through the web interface and the official API. First of all, the official API enables control over a set of parameters, while the web interface does not. For example, ChatGPT API can be set to test at varying temperatures, but the temperature value on ChatGPT Web cannot be controlled. However, while ChatGPT API allows a limited message history, ChatGPT seems to handle an unlimited message history. We used the following evaluation question

[9] platform.openai.com/docs/api-reference/chat

to compare the performance of ChatGPT API and Web:

EQ5: *Does ChatGPT API demonstrate comparable performance to ChatGPT Web in solving WiC tasks?*

Testing ChatGPT API with the MSp strategy would be equivalent to testing it with the FSp strategy due to the limited message history. Thus, we evaluated ChatGPT Web with MSp, aiming to address the following evaluation question:

EQ6: *Can we enhance ChatGPT's performance in WiC tasks by providing it with a larger number of in-context examples?*

To address these evaluation questions, we tested ChatGPT using a single chat for each prompting strategy considered. Since testing ChatGPT Web is extremely time-consuming, we conducted one experiment for each prompting strategy.

Comparing ChatGPT and BERT. The initial introduction of ChatGPT has prompted the belief that ChatGPT is a *jack of all trades* that makes previous technologies somewhat outdated. Drawing upon Kocoń et al. (2023), we believe that, when used for solving downstream tasks as *off-the-shelf* model, ChatGPT is *currently* a *master of none.* It works on a comparable level to the competition, but does not outperform any major SOTA solutions.

By relying on multiple experiments on TempoWiC and HistoWiC, we aimed to empirically assess the potential of ChatGPT for WiC and LSC tasks. In particular, we addressed the following evaluation question:

EQ7: *Does ChatGPT outperform BERT embeddings in detecting semantic change?*

To address EQ7, we evaluated *bert-base-uncased* on TempoWiC and HistoWiC over different layers. Recent research has exhibited better results when utilizing earlier layers rather than the final layers for solving downstream tasks such as WiC (Periti and Dubossarsky, 2023; Ma et al., 2019; Reif et al., 2019; Liang and Shi, 2023). For each layer, we extracted the word embedding for a specific target word w in the context c_1 and c_2. Since the focus of our evaluation was the off-the-shelf use of ChatGPT, we did not fine-tune BERT and simply used the similarity between the embeddings of w in the context c_1 and c_2. In particular, we followed Pilehvar and Camacho-Collados (2019), and trained a threshold-based classifier using the cosine distance between the two embeddings of each pair in the Train set. The training process consisted of selecting the threshold that maximized the performance on the Train set. We trained a distinct threshold-based classifier for each BERT layer and for each WiC task (i.e., TempoWiC and HistoWiC). Then, in our evaluation, we applied these classifiers to evaluate BERT over the TempoWiC and HistoWiC Test sets.

Finally, we addressed the following evaluation question:

EQ8: *Can we rely on the pre-trained knowledge of ChatGPT API to solve the Graded Change Detection (GCD) task?*

Since ChatGPT API has demonstrated awareness of historical lexical semantic change when manually asked about the lexical semantic change of some words (e.g., *plane*), our goal with EQ8 was to automatically test ChatGPT's pre-trained knowledge of historical semantic change covered in the English LSC benchmark. In addressing this evaluation question we relied on the GCD ranking task as defined by Schlechtweg et al. (2018). Thus, we specifically asked ChatGPT to rank the set of 37 target words in the English LSC benchmark according to their degree of change between two time periods, T1 (1810–1860) and T2 (1960–2010). For each temperature, we repeated the same experiment ten times, totaling 110 experiments. Then, for each temperature, we evaluated ChatGPT's performance by computing the Spearman correlation using gold scores derived from human annotation and the average ChatGPT score for each target (see Table 3.5).

Strategy	**Template**
ZSp	Consider the following two time periods and target word. How much has the meaning of the target word changed between the two periods? Rate the lexical semantic change on a scale from 0 to 1. Provide only a score. **Target**: [Target word] **Time period 1**: 1810–1860 **Time period 2**: 1960–2010 **Answer**: [The model is expected to respond with a continuous score s, with $0 \leq s \leq 1$]

Table 3.5: Prompt template for LSC.

Message history. Although one of the many features of ChatGPT is its ability to consider the history of preceding messages within a conversation while responding to new input prompts, ChatGPT API and Web handle message history differently. In ChatGPT API, the message history is limited to a fixed number of tokens (i.e., 4,096 tokens for gpt-3.5-turbo); however, we are not aware of how the message history is handled in ChatGPT Web, where an unlimited number of message for chat seems to be supported.

In our experiments, we use a single chat for each considered prompting strategy, both for ChatGPT API and Web. However, in ChatGPT Web, we considered the full message history for the ZSp, FSp, and MSp strategies. Instead, to avoid exceeding the token limit set by the OpenAI API, we tested ChatGPT API for the ZSp and FSp strategies by considering a message history of 33 messages. Note that due to the token limit, testing the MSp strategy for ChatGPT API wasn't possible, as the limited message history would make MSp equivalent to FSp. The 33-message history was organized as a combination of a *fixed* and a *sliding window*. We set the fixed window to ensure the model is always aware of the task we asked it to answer in the early prompts; instead, we set the sliding window to emulate the flow of the conversation as in ChatGPT Web. In particular, i) in ZSp, the fixed window covers our first prompt (i.e., task explanation) and the ChatGPT answer, while the sliding window covers the i-th prompts and the last 30 messages (i.e., 15 prompts and 15 ChatGPT answers); ii) in FSp, the fixed window covers the first 26 messages (i.e., task explanation and example instances), while the sliding window covers the i-th prompts and the last 6 messages. Figure 3.1

summarizes the message history we set for testing GPT.

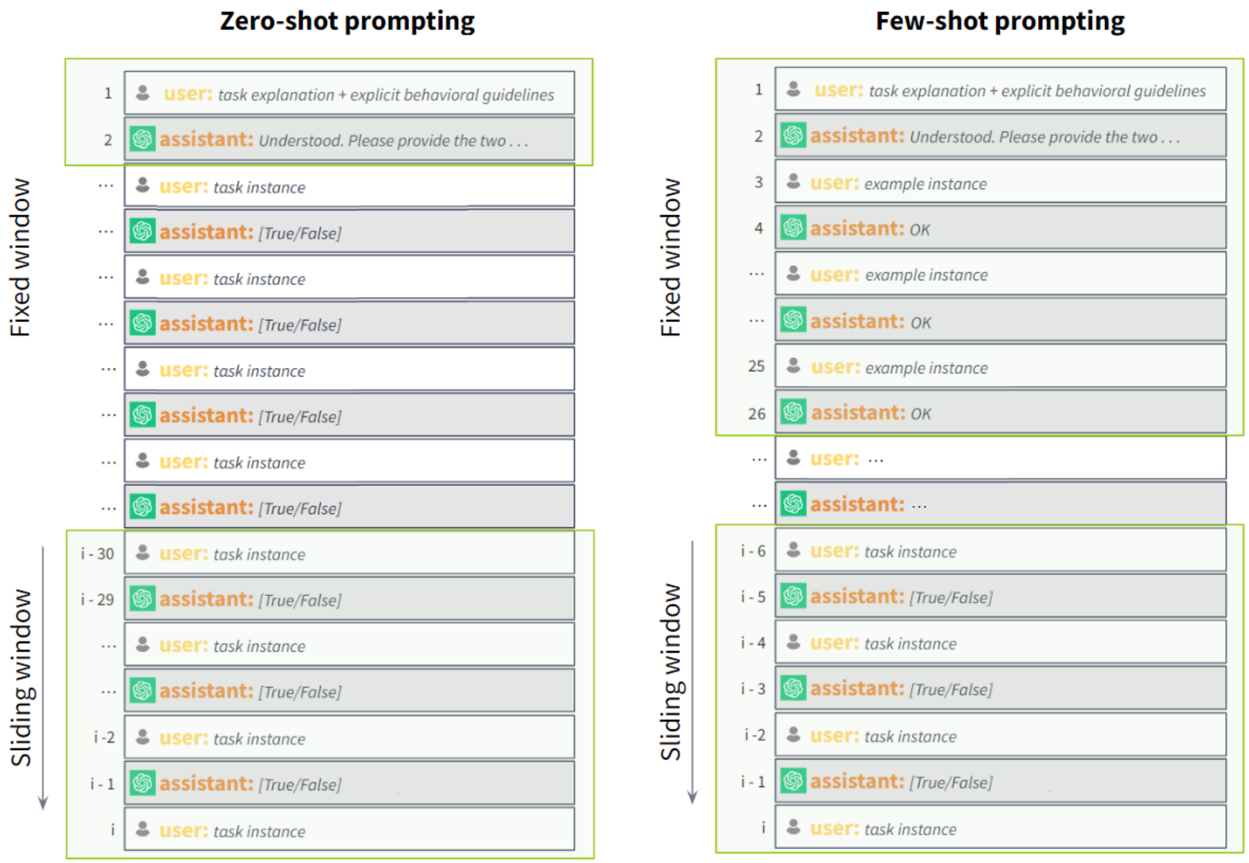

Zero-shot prompting
Fixed window
1 user: task explanation + explicit behavioral guidelines
2 assistant: Understood. Please provide the two . . .
… user: task instance
… assistant: [True/False]
… user: task instance
… assistant: [True/False]
… user: task instance
… assistant: [True/False]
… user: task instance
… assistant: [True/False]
Sliding window
i - 30 user: task instance
i - 29 assistant: [True/False]
… user: task instance
… assistant: [True/False]
i -2 user: task instance
i - 1 assistant: [True/False]
i user: task instance
Few-shot prompting
Fixed window
1 user: task explanation + explicit behavioral guidelines
2 assistant: Understood. Please provide the two . . .
3 user: example instance
4 assistant: OK
… user: example instance
… assistant: OK
25 user: example instance
26 assistant: OK
… user: …
… assistant: …
Sliding window
i - 6 user: task instance
i - 5 assistant: [True/False]
i - 4 user: task instance
i - 3 assistant: [True/False]
i -2 user: task instance
i - 1 assistant: [True/False]
i user: task instance

Figure 3.1: Message history used for ChatGPT API in the zero-shot prompting (ZSp) and few-shot prompting (FSp) strategies. The message history is organized as a combination of a fixed and a sliding window, encompassing a total of 33 messages. The fixed window ensures that the model remains constantly aware of the task we have asked it to address in the initial prompts and the given examples (if any). Conversely, we establish the sliding window to emulate the conversational flow of ChatGPT Web.

3.4 Evaluation results

In this section, we report the results of our experiments, while discussing the findings in regard to each evaluation question.[10]

EQ1: ChatGPT consistently followed our template in nearly all cases, thereby allowing us to evaluate its answers without human intervention. For ChatGPT API, however, we noticed that the higher the temperature, the larger the tendency for deviations from the expected response format (see Figure 3.2). ChatGPT Web only

[10] We provide all our data, code, and results at https://github.com/FrancescoPeriti/ChatGPTvBERT

once answered with an incorrect format. To ensure impartiality, we classified the few ChatGPT responses that did not adhere to the required format as incorrect answers.

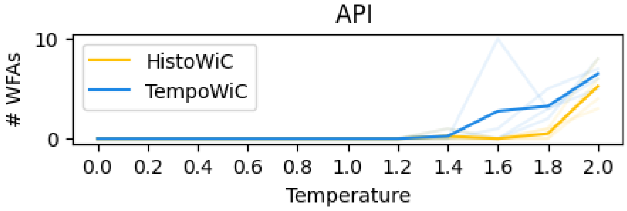


Figure 3.2: Average number of wrongly formatted answers (WFAs) over the temperature values considered. Background lines correspond to each experiment.

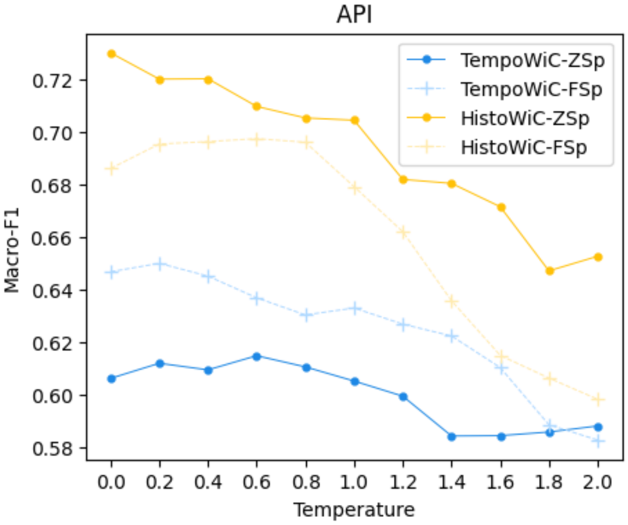


Figure 3.3: Performance of ChatGPT API (Macro-F1) as temperature increases.

EQ2: Figure 3.3 shows the rolling average of the performance of ChatGPT API across different temperatures, prompting strategies, and WiC tasks. By using a window size of 4, we were able to consider 8 different experiments per temperature (for each temperature, we conducted two experiments).[11] Figure 3.4 shows the performance of ChatGPT Web across different prompting strategies and WiC tasks. Further results are reported in Appendix A.

Figure 3.3 and 3.4 show that ZSp consistently outperforms FSp on HistoWiC. By contrast, FSp consistently outperforms ZSp in TempoWiC when the ChatGPT API is used. This result suggests that the in-context learning capability of ChatGPT API is more limited for historical data. In Figure 3.4, ChatGPT's performance with ZSp outperforms that obtained with FSp for both TempoWiC and HistoWiC, although the discrepancy is smaller.

[11] Except for the first and last two temperatures.

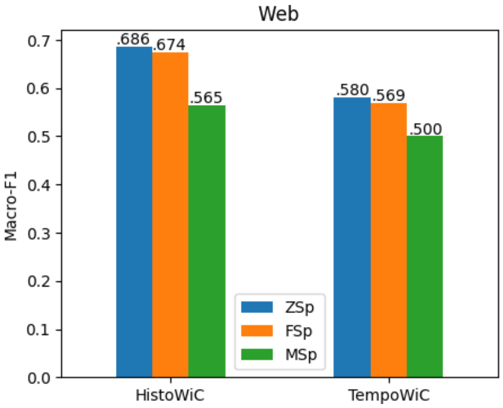


Figure 3.4: Performance of ChatGPT (Macro-F1). Temperature is unknown.

	Macro-F1
Chen et al. (2022b)	.770
Loureiro et al. (2022)	.703
Loureiro et al. (2022)	.670
Lyu et al. (2022)	.625
ChatGPT API	.689
ChatGPT Web	.580
BERT	.743

Table 3.6: Macro-F1 scores obtained by SOTA systems, ChatGPT (best score), and BERT (last layer).

EQ3: Figures 3.3 and 3.4 show that ChatGPT's performance on TempoWiC is consistently lower than its performance on HistoWiC. In particular, in our experiments we observe that ChatGPT's performance ranges from .551 to .689 on TempoWiC and from .552 to .765 on HistoWiC. This suggests that ChatGPT is significantly more effective for long-term change detection than for short-term change detection. For the sake of comparison, we report SOTA performance in Table 3.6. Results from this research are in italics.

EQ4: Figure 3.3 shows that, on average, higher performance is associated with lower temperatures for both TempoWiC and HistoWiC, with accuracy decreasing as temperature values increase. Thus, we argue that high temperatures do not make it easier for ChatGPT API to solve WiC tasks or identify semantic change effectively.

EQ5: ChatGPT Web results are presented in Table 3.7, along with the average performance we obtained through the ChatGPT API across temperature values ranging from 0.0 to 1.0 (API 0–1), from 1.0 to 2.0 (API 1–2), and from 0.0 to 2.0 (API 0–2). As with ChatGPT API, the performance of ChatGPT Web is higher for HistoWiC than for TempoWiC. In addition, our evaluation indicates that ChatGPT Web employs a moderate temperature setting, for we obtained consistent results when using a moderate temperature setting through ChatGPT API. This suggests that the ChatGPT API should be preferred for solving downstream tasks like WiC. It also suggests that the current SOTA evaluations may achieve higher results if the official API were used instead of the web interface. Thus, this implies that previous results using the web interface should be interpreted with caution.

	TempoWiC				**HistoWiC**			
	API	*API*	*API*	*web*	*API*	*API*	*API*	*web*
Temp.	0–1	1–2	0–2	-	0–1	1–2	0–2	-
ZSp	.609	.589	.600	.580	.713	.665	.688	.686
FSp	.636	.606	.622	.569	.693	.626	.657	.674
MSp	-	-	-	.500	-	-	-	.565
all	.622	.598	.611	.550	.703	.645	.672	.642

Table 3.7: Comparison of ChatGPT API and Web performance (Macro-F1).

EQ6: As shown in Figure 3.4, the performance of ChatGPT decreases as the number of example messages increases (from ZSp to MSp). This suggests that improving the performance of ChatGPT requires a more complex training approach than simply providing a few input-output examples. Furthermore, it indicates that the influence of message history is extremely significant in shaping the quality of conversations with ChatGPT. Indeed, a limited message history proved to be beneficial for the evaluation of ChatGPT API through FSp.

EQ7: Figure 3.5 shows Macro-F1 scores obtained on TempoWiC and HistoWiC over the 12 BERT layers (see Table 3.8).

	Layers												
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	**avg**
TempoWiC	.669	.631	.635	.627	.604	.627	.704	.749	.744	.730	.737	.751	.684
HistoWiC	.650	.678	.739	.782	.828	.801	.806	.771	.771	.749	.722	.744	.753

Table 3.8: Comparison of BERT Performance (Macro-F1) for TempoWiC and HistoWiC tasks at different embedding layers.

When considering the final layer, which is conventionally used in downstream tasks, BERT obtains Macro-F1 scores of .750 and .743 for TempoWiC and HistoWiC, respectively. Similar to Periti and Dubossarsky (2023), BERT performs best on HistoWiC when embeddings extracted from middle layers are considered. However, BERT performs best on TempoWiC when embeddings extracted from the last layers are used.

We compared the performance of ChatGPT API and BERT across their respective worst to best scenarios by sorting the Macro-F1 scores obtained by BERT and ChatGPT in ascending order (bottom x-axis). For ChatGPT API, we consider the results obtained through FSp and ZSp prompting for TempoWiC and HistoWiC, respectively. As shown in Figure 3.6, even when considering the best setting, ChatGPT API does not outperform the Macro-F1 score obtained by using the last layer of BERT, marked with a black circle. However, although it exhibits lower performance, the results obtained from ChatGPT API are still comparable to BERT results on HistoWiC when embeddings extracted from the last layer of BERT are used.

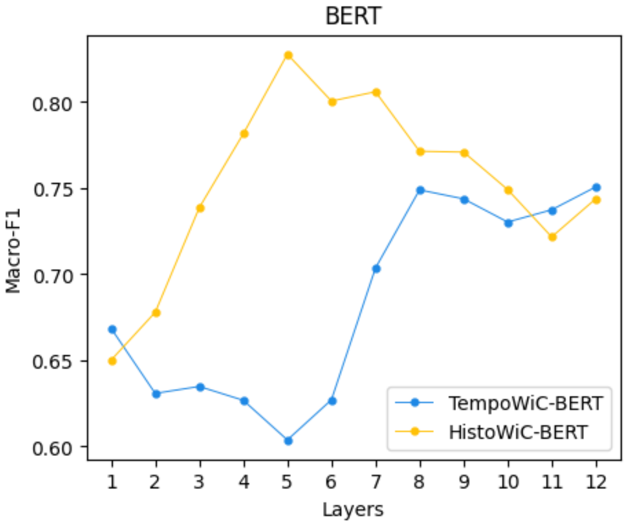


Figure 3.5: Comparison of BERT Performance (Macro-F1) for TempoWiC and HistoWiC tasks across layers

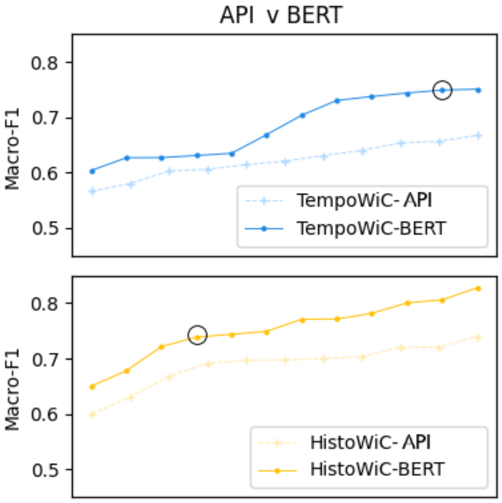


Figure 3.6: ChatGPT API v BERT (Macro-F1). Performance is sorted in ascending order regardless of temperatures and layers. A black circle denotes the use of the last layer of BERT.

Since our goal is to evaluate the potential of ChatGPT for recognizing lexical semantic change, we analyzed the true negative rate and false negative rate scores, because *negative* examples represent semantic change in TempoWiC and HistoWiC datasets. As shown in Figure 3.7, regardless of the temperature and layer considered, ChatGPT falls short in recognizing semantic change for both TempoWiC and HistoWiC compared to BERT. However, it produces fewer false negatives than BERT for TempoWiC.

EQ8: In our experiment, ChatGPT API achieved low Spearman's correlation coefficients for each temperature when ranking the target word of the LSC English benchmark by degree of lexical semantic change.

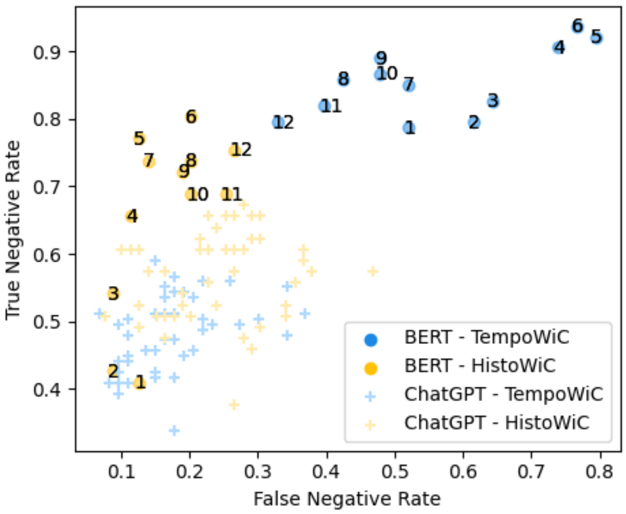


Figure 3.7: True Negative Rate v False Negative Rate. Each cross represents a ChatGPT experiment. Each dot represents the use of a specific layer of BERT.

Higher correlations were achieved by using low temperatures rather than high ones (see Table 3.9).

	Temperature										
	0.0	*0.2*	*0.4*	*0.6*	*0.8*	*1.0*	*1.2*	*1.4*	*1.6*	*1.8*	*2.0*
SemEval-English	.251	.200	.207	.279	.008	.012	.230	.154	.011	.194	.004

Table 3.9: Comparison of ChatGPT API performance (Spearman's correlation) for LSC on SemEval-English at various temperature values.

Table 3.10 shows the ChatGPT API correlation for the temperature 0. For comparison, we report correlations obtained by BERT-based systems that leverage pre-trained models (see Chapter 2 for additional performances). Note that, when BERT is fine-tuned, it generally achieves even higher correlation scores.

	Spearman's correlation
Periti et al. (2024e)	.651
Laicher et al. (2021)	.573
Periti et al. (2022)	.512
Rother et al. (2020)	.512
ChatGPT API	.251

Table 3.10: LSC comparison: correlation obtained by SOTA, *pre-trained* BERT systems and ChatGPT API (temperature=0).

As shown in Table 3.9 and 3.10, the system relying on pre-trained BERT models largely outperforms ChatGPT API, suggesting that an off-the-shell use of ChatGPT is not currently well-adapted for solving LSC downstream tasks.

BERT for Semantic Change Detection. There are notable differences between the Macro-F1 for TempoWiC and HistoWiC in terms of how the results increase and decrease across layers (see Figure 3.5). For TempoWiC the results increase until the **8th layer**,[12] after which they remain almost stable. Conversely, for HistoWiC the BERT performance rapidly increases until the 5th layer, after which it linearly decreases until the 12th layer. As regards Tempo WiC, we hypothesize that BERT is already aware of the set of word meanings considered for evaluation as it was pre-trained on modern and contemporary texts. As regards HistoWiC, we hypothesize that BERT is not completely aware of the set of word meanings considered for evaluation and that word representations adopted for the historical context of HistoWiC[13] might be slightly tuned. Thus, using medium embedding layers could prove beneficial in detecting semantic change, as these layers are less affected by contextualization (Ethayarajh, 2019). In other words, for HistoWiC, we hypothesize that the performance diminishes in the later layers due to the increasing contextualization of the medium and final embedding layers, which reduces the presence of noise in untuned word representations. This prompts us to question the appropriateness of using the last embedding layers to recognize historical lexical semantic change. *We will address this question in Chapter 7.*

3.5 Discussion and considerations

In the study presented in this chapter, we empirically investigated the capability of ChatGPT to detect *semantic change*. We used the TempoWiC benchmark to assess ChatGPT to detect short-term semantic change, and introduced a novel benchmark, HistoWiC, to assess ChatGPT's ability to recognize long-term change. When considering the standard 12 layer of BERT, our experiments show that ChatGPT achieves comparable performance to BERT (although slightly lower) in regard to detecting long-term change, but performs significantly worse in regard to recognizing short-term change. We find that BERT's contextualized embeddings consistently provide a more effective and robust solution for capturing both short- and long-term change in word meanings.

There are two possible explanations for the discrepancy in ChatGPT's performance between TempoWiC and HistoWiC: i) HistoWiC might involve word meanings not explicitly covered during training, potentially aiding ChatGPT in detecting anomalies; ii) TempoWiC involves patterns typical of Twitter (now X), such as abbreviations, mentions, or tags, which may render it more challenging than HistoWiC.

However, there are limitations we had to consider in the making of this evaluation. Firstly, a limitation arises when working with temporal HistoWiC benchmarks. While we ensure the utilization of diachronic data, we cannot guarantee that if the meaning of a word differs across contexts, it unequivocally indicates either the presence of stable polysemy (existing stable multiple meanings) or exemplifies a semantic change (either a new sense that it did not previously possess or a lost sense that it no longer has).

Other limitations are about the use of language models. We could not evaluate ChatGPT across different languages due to both price and API limitations. This means that while the results hold for English, we do

[12] *We will observe similar results in Chapter 7.*

[13] 1810–1860, as referenced in Schlechtweg et al. (2020).

not know how ChatGPT will behave for the other languages. Although we are aware of recent open-source solution such as LLaMA, it still necessitates expensive research infrastructure, and we thus chose to focus on ChatGPT. *We will investigate LLaMA in Chapter 8 and 9.*

Like all research on ChatGPT (Laskar et al., 2023; Kocoń et al., 2023; Zhong et al., 2023), our work has a significant limitation that we cannot address: our ChatGPT results are not entirely reproducible as ChatGPT is inherently nondeterministic. In addition, like Zhong et al. (2023) and Jiao et al. (2023), we found that time and economic constraints when using ChatGPT dictated that our evaluation of the software had to be based on only a subset of the TempoWiC and HistoWiC dataset.

In our study, we utilized ChatGPT-3.5. This could be considered a limitation, given the availability of its foundational or more recent chat versions. However, we opted for ChatGPT instead of its foundational version as it has already undergone instruction tuning. In addition, we chose ChatGPT-3.5 based on the guidance provided in the OpenAI documentation at the time of this study.[14] Additionally, we argue that ChatGPT-3.5 is a cheaper alternative than the current models, making the investigation of ChatGPT-3.5 still significant for researchers with limited economic resources. We acknowledge that OpenAI continues to train and release new models, which could potentially affect the reproducibility of our results.

One of the many features of ChatGPT is its ability to incorporate the history of preceding messages within a conversation while responding to new input prompts. However, there remain several unanswered questions regarding how this history influences the model's answers. This holds true even for the zero-shot prompting strategy, where a general setting is lacking. Multiple prompts can be provided as part of the same chat or across different chats. For simplicity, and similar to previous research, we assigned only one chat for each ZSp experiment.

Finally, as highlighted by Laskar et al. (2023), since the instruction-tuning datasets of OpenAI models are unknown (that is, not open source), the datasets used for evaluation may or may not be part of the instruction-tuning training data of OpenAI. Additionally, Balloccu et al. (2024) raised concerns about indirect data leaking due to models being iteratively improved using data from users.

Despite these limitations, we argue that our work is significant as it may prompt new discussion on the use of LLMs such as BERT and ChatGPT, while also dispelling the expanding belief that the use of ChatGPT as *off-the-shelf* model *already* makes BERT an outdated technology.

Nonetheless, during the course of our research, updates to ChatGPT became available and gained popularity, leading researchers and practitioners to conduct new experiments on these updated models. Particularly noteworthy is a recent study by Karjus (2023), which showcased remarkable performance on LSC using the GPT-4 model. Inspired by this research, we focused on further exploring the capabilities of GPT-4 for modeling semantic change and word meaning in context. Our results indicate that GPT-4 is more powerful than GPT-3. However, the mentioned limitations still apply and must be considered when interpreting our results. *We will further investigate the use of GPT-4 in Chapter 7.*

[14] https://platform.openai.com/docs/guides/gpt/which-model-should-i-use

Chapter 4

Extending the modeling to multiple time periods

> "*Whilst this planet has gone cycling on according to the fixed law of gravity, from so simple a beginning endless forms most beautiful and most wonderful have been, and are being, evolved.*"
>
> Charles Darwin, *On the Origin of Species*

4.1 Introduction

Since the LSC shared tasks proposed at SemEval-2020 (Schlechtweg et al., 2020), there is an established evaluation framework for LSC to compare the performance of various models and approaches. However, given the substantial annotation efforts required to create reliable benchmarks over multiple time periods, the framework is typically adopted to create simplified benchmarks over two time periods, with gold labels for semantic change without diachronic sense labels (Ling et al., 2023; Chen et al., 2023a; Kutuzov et al., 2022a; Zamora-Reina et al., 2022b; Kutuzov and Pivovarova, 2021c; Basile et al., 2020).[1] With such benchmarks, the research community has focused its efforts on a simplified modeling of semantic change between *two time periods*. We reviewed this simplified view of LSC in Chapter 2. However, while this view has served as a foundational block of modeling, we believe that more comprehensive efforts are crucial to address research questions posed in the humanities and social sciences over *multiple time periods*.

Conceptually, the LSC problem implicitly involves a fundamental step of *diachronic word sense induction* to distinguish each individual sense of a word over all the *multiple time periods* of interest (Periti et al., 2024e; Alsulaimani and Moreau, 2023; Alsulaimani et al., 2020; Emms and Jayapal, 2016). However, the

[1] Kutuzov and Pivovarova (2021c) introduced a benchmark encompassing two time intervals. However, these intervals have been treated independently, leading to their consideration as two distinct sub-benchmarks over a single time interval.

computational challenges in handling large corpora and the absence of comprehensive benchmarks have in practice led to a simplified modeling focused on *two* time periods t_1 and t_2 only. These are either modeled *separately* t_1, t_2 or in a single time interval $\langle t_1, t_2 \rangle$ considering all the data *jointly*.

Typically, approaches over two time periods are assumed to be directly extendable to real scenarios involving multiple time periods. For example, approaches designed for a single interval $\langle t_1, t_2 \rangle$, can be iteratively re-executed across multiple, contiguous intervals $\langle t_1, t_2 \rangle$, $\langle t_2, t_3 \rangle$, ..., $\langle t_{n-1}, t_n \rangle$ (Giulianelli et al., 2020). However, multiple re-executions present a computational challenge that significantly escalates as the number of considered periods increases. Procedures that were initially considered optional steps to expedite modeling in two time periods become fundamental over multiple time periods. For instance, since words can occur thousands of times in a diachronic corpus, it becomes imperative to randomly sample a limited number of occurrences and to leverage hardware components, such as GPU processor units.

Due to the absence of diachronic lexicographic resources (e.g., dictionaries, thesauri), and the gap between a general resource and specific data, the modeling of word sense is commonly approached in an *unsupervised* manner. Clustering techniques are generally employed to aggregate usages of a specific word into clusters, with the idea that each cluster denotes a specific word meaning that can be recognized in the considered documents. However, clusters of usages (regardless of method of clustering) do not necessarily correspond to precise senses (Martinc et al., 2020b), but typically represent noisy projections related to specific context (Kutuzov et al., 2022b). As a result, manual activity is always required to translate the automatically derived clusters into a *diachronic sense inventory*. This sense inventory is the basis for interpreting the identified semantic change and modeling sense evolution (see Figure 4.1). While automatic methods, such as keywords extraction (Kellert and Mahmud Uz Zaman, 2022), or generating definitions for word usages (Giulianelli et al., 2023), have been proposed to support cluster interpretation, a reliable interpretation still needs manual supervision. Therefore, when multiple time periods are considered, interpretability challenges increase several orders of magnitude, making the direct re-execution of existing approaches unsuitable for effectively detecting semantic change.

We thus argue that the *diachronic word sense induction* over *multiple time periods* inherent to LSC requires more careful considerations compared to the simplified modeling over *two time periods*. More efforts should be devoted to develop approaches for assisting text-based researchers like linguists, historians and lexicographers as much as possible.

Chapter outline.

This chapter includes materials originally published in the following publication, which is currently under review:

> Francesco Periti and Nina Tahmasebi. 2024**b**. Towards a Complete Solution to Lexical Semantic Change: an Extension to Multiple Time Periods and Diachronic Word Sense Induction. In Proceedings of the 5th Workshop on Computational Approaches to Historical Language Change, pages 108–119, Bangkok, Thailand. Association for Computational Linguistics.

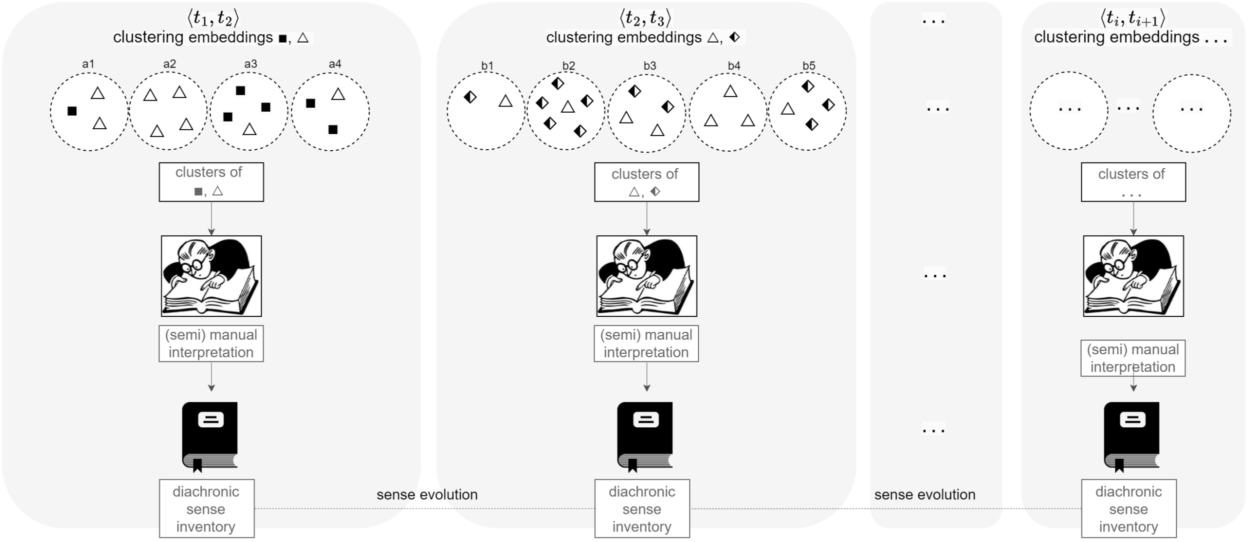

⟨t1, t2⟩
clustering embeddings ■, △
a1
a2
a3
a4
clusters of ■, △
(semi) manual interpretation
diachronic sense inventory
sense evolution
⟨t2, t3⟩
clustering embeddings △, ⬖
b1
b2
b3
b4
b5
clusters of △, ⬖
(semi) manual interpretation
diachronic sense inventory
sense evolution
⟨ti, ti+1⟩
clustering embeddings . . .
clusters of . . .
(semi) manual interpretation
diachronic sense inventory

Figure 4.1: Word usages and their corresponding representations, for time period t_1, t_2, and t_3 are denoted with ■, △, ⬖, respectively. Typically, the clustering of representations is done for individual time intervals (i.e., two time periods jointly) and manual supervision is required to translate the clusters of each time interval to a diachronic sense inventory. The amount of manual supervision increases with the number of considered time intervals.

In this chapter, we discuss the complexities inherent in modeling semantic change for each word sense individually over multiple time periods. We challenge the general assumption that conventional approaches designed to address LSC over two time periods are easily extendable over multiple time periods. Currently, contextualized embeddings represent the preferred tool for addressing LSC; hence, we will use these as an example. Our discussions, however, are more general and can be applied regardless of which model is used to represent individual word usages – such definitions (Giulianelli et al., 2023), co-occurrence vectors (Schütze, 1998), or bag-of-substitutes (Kudisov and Arefyev, 2022; Arefyev and Zhikov, 2020) – or sense clusters in general, as presented in Tahmasebi and Risse (2017). Specifically, in this chapter, we advocate for an alternative modeling of LSC over multiple time periods and discuss significant implications for both computational modeling and the creation of benchmarks.

The chapter is organized as follows. In Section 4.2, we extend the state-of-the-art presented in Chapter 2 by further discussing the modeling of word senses through clustering. In Section 4.3, we address the limitations of the current LSC and propose five distinct approaches to trace semantic change and the evolution of word meanings. Additionally, in Section 4.4, we outline three distinct settings for assessing semantic change over multiple time periods. Finally, in Section 4.5, we discuss relevant considerations for modeling LSC.

4.2 Modeling senses through clusters

The clustering of representations via word sense induction serves as a tool to operationalize word senses in an unsupervised fashion through unstructured text (Lake and Murphy, 2023). On one hand, this *operationalization* offers a flexible adaptation to the data under consideration and allows to derive senses that do not necessarily need to be aligned with available static lexicographic resources (Kilgarriff, 1997). For instance, senses derived from youth slang (Keidar et al., 2022), or scientific texts are unlikely to align with a general lexicon meant to cover the whole spectrum of a given language.

On the other hand, as computational models derive information from the contexts surrounding word tokens, sense modeling tends to emphasize word usages rather than word meanings (Tahmasebi and Dubossarsky, 2023; Kutuzov et al., 2022b). Thus, while ideally we would like each cluster to correspond to one, and only one sense, in practice, multiple clusters may correspond to different nuances of the same sense. This effect is further amplified when considering data from diverse time, domains, or genres, where distinct linguistic registers, styles, or co-occurrence patterns may result in different senses.

Additionally, the interpretation of clusters as senses requires a notion of (word) "meaning" that can both differ in the mind of humans according to social or cultural background and age, as well as in the varying usages of a word in context. Thus, the mapping of *clusters* to *senses* involves i) identifying commonalities on the usages of each cluster that may be judged differently, as well as ii) mapping these commonalities to word meanings. The outcome results in a *sense inventory.*

4.2.1 Approaches to LSC over multiple time periods

Modeling LSC involves computationally deriving word senses progressively over time. This entails re-executing the following steps multiple times:

1) extraction of the word occurrences from both t_1 and t_2;

2) computational representation of each occurrence (the current standard is to leverage pre-trained contextualized embeddings);

3) word sense induction by aggregating embeddings with a clustering algorithm;

4) assessment of semantic change by leveraging a distance measure on the embeddings from t_1 and t_2.

When *form-based* are employed, individual senses are not induced **(3)**, thus there is no easy way to discern individual senses from the change score without integrating "close reading" by humans. Sense-based approaches remedy this by relying on all steps **(1-4)** but generally induce senses **(3)** in a *synchronic* way, without considering the temporal nature of the documents (Ma et al., 2024a; Periti et al., 2022). That is, they consider all the documents from t_1 and t_2 available as a whole and perform a single clustering activity over the entire set of generated embeddings, regardless of their time origin.

At each execution i, a set of clusters is generated and humans are needed to identify and update the sense inventory. This involves mapping the clusters generated at the i-th execution to senses and aligning senses temporally (see Figure 4.1).

The way senses align over time gives us important insights into how word meanings change. Classifying *types* of semantic change has been long studied and different schema have been proposed (Blank, 1997; Ullmann, 1957; Bloomfield, 1933; Stern, 1931; Bréal, 1904; Darmesteter, 1893; Paul, 1880; Reisig, 1839). Among others, common types of change include:

1. *broadening*: when the meaning of a word becomes more inclusive or general over time. For example, `dog` was used to refer not to any old dog, but to some specific large and strong breeds;
2. *narrowing*: when the meaning of a word becomes more specific or limited over time. For example, `girl` was used to refer to people of either gender;
3. *novel senses*: when entirely new meanings or senses of a word emerge over time. For example, `rock` as a music genre;
4. *metaphorical* extensions: when a word's meaning is extended metaphorically to represent something different from its original sense. For example, the use of `surfing` web searches.

The result is a *diachronic* sense inventory with temporal information on the active senses at each time, as well as potential relationships between senses.

To facilitate the interpretation of semantic change and the evolution of word meaning, the current, *synchronic* modeling of senses can benefit from *diachronic* modeling encompassing both incremental word sense induction and cluster alignment (Kanjirangat et al., 2020). Aligning clusters computationally will allow the simultaneous interpretation of multiple clusters, thereby reducing the burden of manual supervision at each time period. Clusters aligned over time can potentially suggest the continuation of an active sense, as well as the broadening and narrowing of meanings. In contrast, clusters not aligned over time can reveal both the continuation of different senses, as well as types of substantial change, like metaphoric extension.

Thus far, word meanings have been modeled through conventional clustering algorithms such as Affinity Propagation (Martinc et al., 2020b) or K-Means (Kobayashi et al., 2021). However, these algorithms were originally designed for one-time data clustering and are not inherently suited to handle temporal dynamics. Specifically, clusters generated at t_{i-1} can become mixed up when re-executing the algorithm with both previous data and new data points at time $\langle t_{i-1}, t_i \rangle$. Consequently, objects that were previously clustered together at time t_{i-1} may either remain in the same cluster or be reassigned to different clusters based on the updated data at time t_i. This dynamic nature complicates the task of tracking the history of specific clusters across different time periods, and can lead to the creation of noisy clusters, especially when new data points arrive according to a skewed distribution.

4.2.2 Diachronic sense clustering.

Conventional unsupervised clustering algorithms do not incorporate the faithfulness properties typical in *incremental clustering* literature, where clustering activities at any given point in time should remain faithful to the already existing clusters as much as possible (Chakrabarti et al., 2006) while at the same time be flexible to fit the new data. This would avoid dramatic change in clusters from one time-step to the next that do not derive from semantic change, but from differences in the underlying documents over time.

To this end, we argue that, for each target word, modeling LSC over time should involve *monitoring* the evolution of each individual sense across all the time periods under consideration, as well as *tracing* the types of each change. However, this extension is not straightforward; instead, it requires crucial time series analysis to mitigate potential noise introduced by the predictions of computational approaches (Kulkarni et al., 2015).

Monitoring and tracing word meaning evolution and semantic change require careful consideration in the current *four-step pipeline* of sense-based approaches. As for scalability and interpretability issues related to **(1-3)**, suggestions and workaround are discussed in Periti and Montanelli (2024) and Montariol et al. (2021). We further discuss the extension of steps **(3)** and **(4)** when considering multiple time points. In particular, we discuss *diachronic word sense induction* in Section 4.3, and *semantic change assessment* in Section 4.4.

4.3 Diachronic word sense induction

For the sake of simplicity, consider a diachronic corpus C spanning three general, consecutive time periods t_1, t_2, t_3, not necessarily contiguous. This simplification does not lead to any loss of information, but serves to aid the discussion in a clear and concise fashion. At the same time, three time points are easily extendable to the general case of tens or hundreds of time periods. Word usages, and their corresponding representations (i.e., contextualized embeddings), for time period t_1, t_2, and t_3 are denoted with ■, △, ⬖, respectively. In the following, we present five different approaches for monitoring the evolution of word meanings and discuss suitability, and drawbacks.

4.3.1 Clustering over consecutive time intervals

Clustering algorithms used for *jointly* modeling senses over two time periods t_1 and t_2 can be progressively re-executed over consecutive pairs of time periods $\langle t_1, t_2 \rangle$ and $\langle t_2, t_3 \rangle$. To facilitate the interpretation of sense evolution, a cluster alignment step is thus required between consecutive re-executions. For instance, in Figure 4.2, the clusters generated in step (B) are linked to those generated in step (A) through a cluster alignment step (C) (Deng et al., 2019).

When clustering over consecutive time intervals $\langle t_1, t_2 \rangle, \ldots, \langle t_{n-1}, t_n \rangle$, the embeddings from $n - 2$ time periods (all time periods but first and last) are clustered twice. For instance, consider the embeddings △ from t_2 in Figure 4.2: (A) they are first clustered with the embeddings ■ from t_1, and (B) then re-clustered with the embeddings ⬖ from t_3. When a limited number of word usages is available, this approach can potentially enhance the emergence of certain senses, as patterns of embeddings from t_{i-1} are reinforced by additional evidence (if present) from t_i. However, this compromises the faithfulness property, as embeddings from t_i can be clustered differently when considered jointly with t_{i-1} compared to when considered jointly with t_{i+1} (from a past and future perspective respectively).

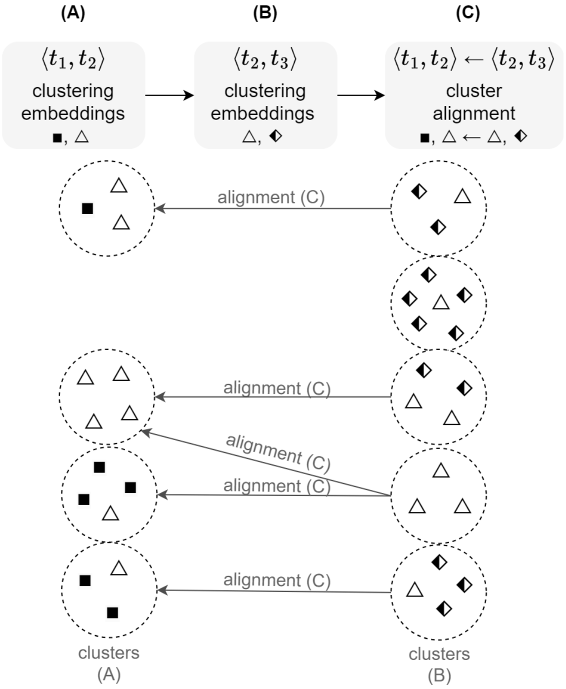


Figure 4.2: Clustering over consecutive time intervals.

4.3.2 Clustering over consecutive time periods

When a substantial number of documents is available for each time period, there is no need to cluster the embeddings of a time *interval* as a whole. Instead, the embeddings of each time *period* can be clustered individually, and a cluster alignment algorithm can be applied progressively to link the clusters across time periods (Kanjirangat et al., 2020; Montariol et al., 2021). This approach is represented in Figure 4.3. Step (A), (B), and (D) represents the application of a conventional clustering algorithm over the embeddings of time period t_1, t_2, t_3, respectively. Step (C) and (E) represent cluster alignment steps to link the clusters generated through step (B) to the cluster generated through step (A), and in turn, the clusters generated through step (D) to the cluster generated through step (B) (Deng et al., 2019).

Clustering over time periods involves a similar number of clustering activities and cluster alignment steps as clustering over time intervals. However, each clustering activity is more scalable, as it involves a smaller number of embeddings.

4.3.3 One-time clustering over all time periods

Embeddings from all the considered time periods can be clustered jointly in one single execution. For instance, in Figure 4.4 step (A), embeddings ■, △, ⬖ are clustered together as a whole. This single clustering activity results in clusters that may include embeddings from various combinations of time periods. For example, a cluster may include embeddings from a single, all, or selected time periods. A cluster alignment step (B) can be further executed to enable the modeling of sense evolution and change type.

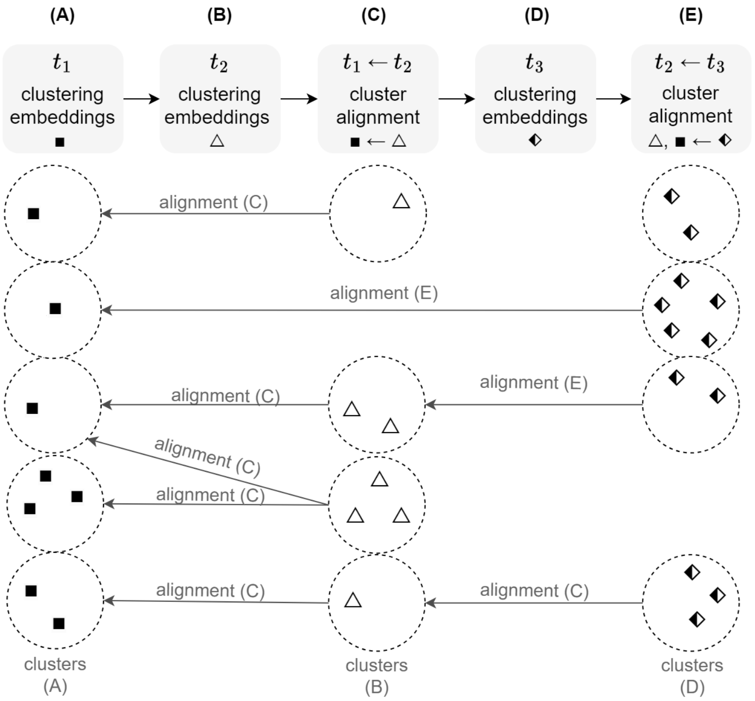

(A)
(B)
(C)
(D)
(E)
t_1 clustering embeddings ■
t_2 clustering embeddings △
$t_1 \leftarrow t_2$ cluster alignment ■ ← △
t_3 clustering embeddings ◆
$t_2 \leftarrow t_3$ cluster alignment △, ■ ← ◆
alignment (C)
alignment (E)
alignment (C)
alignment (E)
alignment (C)
alignment (C)
alignment (C)
alignment (C)
clusters (A)
clusters (B)
clusters (D)

Figure 4.3: Clustering over consecutive time periods.

When dealing with hundreds of time periods and a significant number of embeddings at once, clustering can be unfeasible due to scalability issues. In real scenarios, a diachronic corpus can be *dynamic* (Castano et al., 2024; Periti et al., 2024e, 2022), where documents from subsequent time periods are not available as a whole but are progressively added (e.g., *posts* from social networks, Kellert and Mahmud Uz Zaman, 2022; Noble et al., 2021). In such scenarios, this approach is thus not suitable as it would require re-execution of the clustering from scratch when new documents are added.

Furthermore, the use of conventional clustering algorithms is generally insensitive to the order of time periods, allowing embeddings of later time periods to influence the patterns of the earlier time periods. This risks leading to a global view of word meaning while precluding a local view where smaller and gradual variations of individual senses as well as small sense clusters are missed. These issues can be mitigated by considering the temporal order of documents in the clustering activity (Smyth, 1996).

4.3.4 Incremental clustering over time periods

Incremental clustering algorithms are designed to effectively address the temporal nature of data (Kulkarni and Mulay, 2013). These algorithms operate under the assumption that objects arrive progressively, and clustering is performed incrementally as new data becomes available. Thus, they are a suitable option to model the dynamic nature of language where temporal progression is key. When employed for diachronic word sense induction, they can efficiently and directly update the prior clustering results by processing and assimilating new data into existing clusters. The word usages observed in past time periods are consolidated

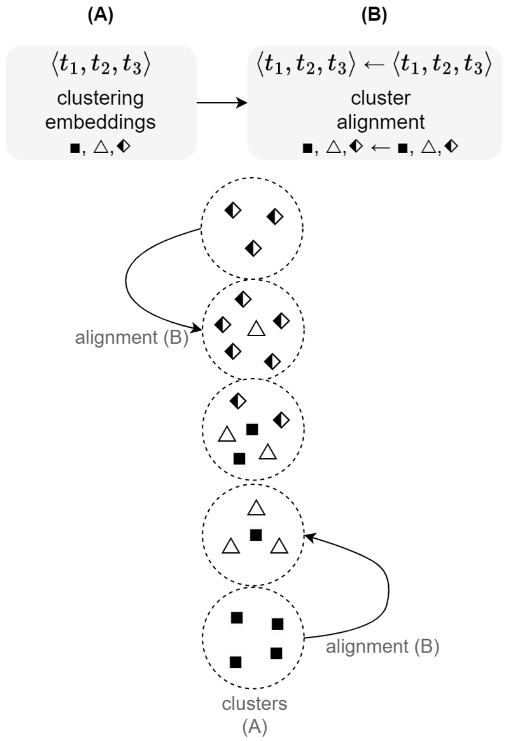


Figure 4.4: One-time clustering over all time periods.

into a set of clusters that constitute the *memory* of the word meanings observed thus far (Periti et al., 2022). This memory then serves as a foundation for understanding subsequent word usages in the current time period. Like Figure 4.4, Figure 4.5 represents similar steps (A-C) without alignment as clusters generated in step (A-C) are directly and consecutively updated.

Some of the incremental algorithms implement the faithfulness property in an *evolutionary* way: once a cluster has been created, it can only gain new members (i.e, word usages) but can never lose any members that have already been assigned to it. Meanwhile, the word usages observed in the present must be stratified or integrated over those from the past, that is, either be placed in existing clusters, or create new clusters. Other algorithms implement the faithfulness property in a more flexible way and enable small changes in past clusters when more evidence is available.

4.3.5 Scaling up with form-based approaches

Regardless of the complexity of each presented method, it is difficult to scale an approach to the level of whole vocabulary in a large corpus. In addition, some senses remain stable for a long time before they potentially change meaning, others never change. Therefore, clustering the senses during the stability periods of words is superfluous. To reduce computational needs and scale to the entire vocabulary, form-based approaches (without sense-induction) can be used to monitor stability allowing the use of more powerful sense-based approaches only when there is indication of change.

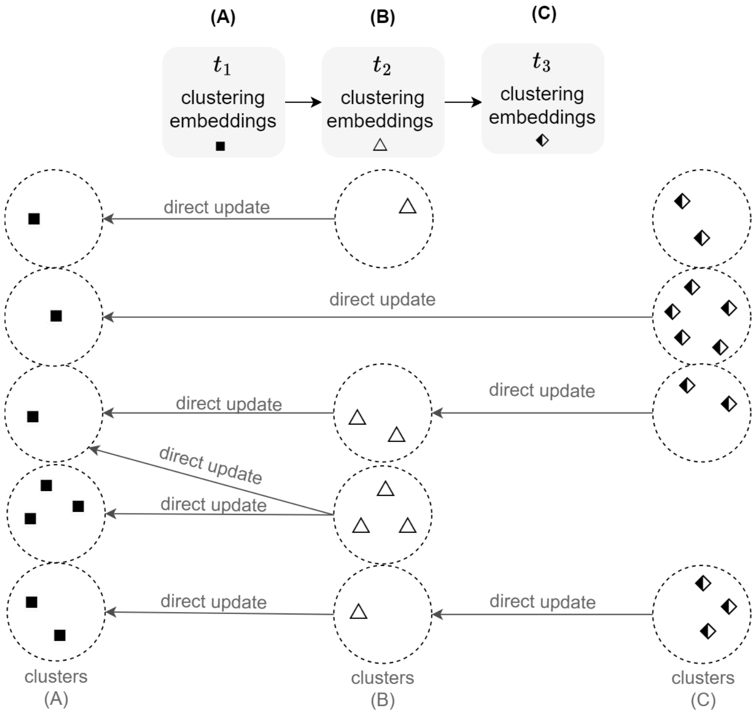


Figure 4.5: Incremental clustering over time periods.

By considering change only in the general usage of a word, form-based approaches reduce the semantic change problem significantly. Thus, they serve for two important purposes: first, they can be used to quantify the degree of change at the vocabulary level, and thus give us the opportunity to quantify change during different time periods (e.g., before and after WWI v. WWII); secondly, they can be used to find words and periods of interest.

Such a kind of stability monitoring can be done via change point detection (Kulkarni et al., 2015) and be integrated with diachronic sense modeling as shown in Figure 4.6. In particular, step A involves quantifying semantic change through form-based assessment to detect change points across the entire time span covered by the corpus. Step B involves modeling each individual sense of the word around the detected change point(s) through approaches presented in Section 4.3.1-4.3.4.

4.4 Semantic change assessment

The diachronic word sense induction is independent from the assessment of change at the level of senses or words. While the modeling of word meaning relies on the notion of word senses, the assessment of change depends on the research questions that we want to investigate. E.g., considering a perfect sense inventory we may want to ask how many meanings have been lost and gained, and if change is more evident in some time intervals compared to others. The answer to these depends on the way we assess change.

Assessment of change, like sense induction, has focused on two time intervals which is the smallest unit

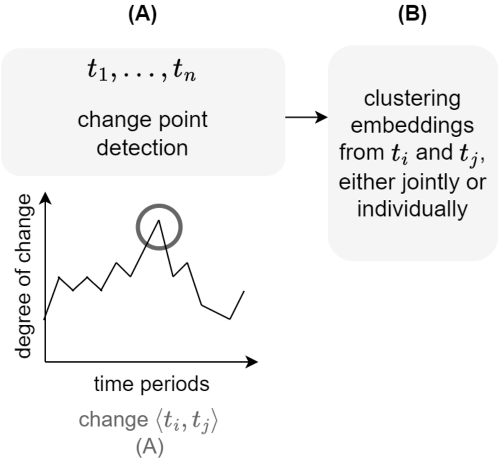

(A)
(B)
$t_1, \ldots, t_n$
change point detection
clustering embeddings from t_i and t_j, either jointly or individually
degree of change
time periods
change $\langle t_i, t_j \rangle$
(A)

Figure 4.6: Scaling up with form-based approaches.

over which we can quantify change. However, generalizing from two intervals to multiple intervals is not trivial and needs considerations that depend heavily on the kind of research question that is being asked, as well as the kind of data available. Short-term data versus long-term data, or small contra large data require different strategies for quantifying change. Here we present some possible strategies that extend to multiple time periods.

4.4.1 Assessment over consecutive time intervals

represents a general way to assess semantic change over time $\langle t_1, t_2 \rangle$, $\langle t_2, t_3 \rangle$, …, $\langle t_{n-1}, t_n \rangle$. This kind of assessment can be affected by i) (random) fluctuations in the underlying corpus, where the coverage of topics can be heavily influenced by real-life events; and ii) noisy artifacts of the computational modeling, e.g., influenced by frequency. The use of time series analysis or statistical tests can reduce the effect of potential artifacts from the data and capture only significant changes evident in the time series (Liu et al., 2021b; Kulkarni et al., 2015).

This assessment represents a useful solution for scenarios where the focus is on detecting immediate changes, such as in rapidly evolving fields or during specific events that might impact language usage. When comparing $\langle t_{i-1}, t_i \rangle$, the assumption is that all the active word meanings in t_i, except for the new or changed ones, are active also in t_{i-1}. However, some senses are periodic and an undesirable side-effect is that they may be detected as change each time they appear and disappear as they are not represented in t_{i-1}.

4.4.2 Pairwise assessment over time periods

Sometimes research questions may be tailored to specific time intervals (e.g, *before* and *after* the time period t_i of the corona pandemic). Thus, this assessment aims to quantify the change across specific time intervals $\langle t_{i-1}, t_i \rangle$ and $\langle t_j, t_{j+1} \rangle$ such that $i < j$. This assessment is also useful for identifying changes in periodic senses when the periodicity of the sense is known. For example, the meaning of the term *gold* is related to the Olympic games that take place every fourth year.

This assessment is also useful when research questions are tailored to a specific type of change irrespectively when the change occurs. For example, when a diachronic sense inventory is available, broadening or narrowing can be investigated regardless of their time-specific appearance.

When all possible time intervals are considered, this assessment is associated with a computational complexity of $\mathcal{O}(n^2)$ where n is the number of considered periods. However, it provides a broader view of how meaning evolves over different spans, capturing trends that may not be apparent in consecutive intervals. For example, gradual changes over time would not appear with assessment over consecutive time intervals as too little evidence would be present, but will appear as radical changes with larger gaps between intervals.

By considering all the possible time intervals is also possible to quantify the **global** level of change over the whole corpus. This method is insensitive to the order of the time periods and is useful for capturing overarching trends and patterns in semantic change across the entire timeline.

4.4.3 Cumulative assessment over time

When research questions focus on the novel senses gained at time period t_i, the comprehensive overview of active sense from the past must be considered $\bigcup_{j=1}^{i-1} t_j$. Instead of considering only consecutive or specific time intervals, each new time period should be compared with the full diachronic sense inventory. Cumulative assessment emphasizes the overall evolution of meaning, providing a holistic view of changes from the beginning to the end of the timeline. It is useful for consolidating the evidence across multiple time periods which would not suffice on their own. For example, when research questions focus on the novelty introduced in time period t_i compared to the past periods, the assessment of change should consider the cumulative evidence of the past as a single, large time period. A similar assessment can be employed when research questions want to compare a past time period t_i with respect to the following $\bigcup_{j=i+1}^{n-1} t_j$.

4.5 Discussion and considerations

Computational modeling of semantic change has long been done in a simplified way due to the challenges related to modeling senses across multiple time periods. However, sense inventories and the type of change a word exhibits, are fundamental aspects for text-based researchers like historians, linguists and lexicographers, and therefore, the full complexity of semantic change must be taken into consideration in the computational modeling. Now that we have powerful language models like XL-LEXEME (Cassotti et al., 2023a) and GPT (OpenAI, 2023) there are no excuses for taking a simplistic view on the modeling of semantic change.

In this chapter, we have presented possible extensions to expand on the simplistic view. These extensions have equal implications both for the computational modeling as for the generation of manually annotated benchmarks which has also been done over two time periods due to the sheer volume of required annotations.

Crucial for the usefulness of semantic change studies is a *diachronic sense inventory* where the different senses are linked together to capture semantic change type and linguistic relation. It is by using the diachronic sense inventory that the majority of the research questions can be answered. These pertain both to linguistic, language-level questions, but also to societal and cultural enquiries where text can be used as evidence. How to best frame and store the diachronic sense inventory is still an open issue and requires involvement from the communities around computational modeling of semantic change, word sense induction and lexical semantics in general, as well as the text-based researchers that will use the outcome.

Human supervision is necessary to develop a reliable sense inventory. As diachronic corpora can span multiple time periods and contain millions of documents, automatic supervision support is mandatory to reduce manual efforts as much as possible. In this regard, aligning similar clusters and detecting change types to speed up the interpretation process is as crucial as it is difficult. Employing different kinds of diachronic word sense induction and assessment as outlined here, will lead to different amounts of manual interaction.

Aligning clusters over time poses a very challenging task, as some clusters may represent outliers, time intervals may be characterized by different numbers of clusters, and multiple noisy (or nuanced) clusters denoting the same meaning may emerge. As a result, the cluster alignment often involves the discretization of a fuzzy problem (Kianmehr et al., 2010), that is the creation of new global clusters that encompass sets of fuzzy clusters. Furthermore, when clusters are aligned through a posteriori step rather than being linked and updated directly, the alignment process (worst case) involves comparing each cluster with every other cluster across all time periods. This risks amplifying the potential level of noise and requires intricate decisions typically taken without any theoretical basis.

Thus far, the research community has focused more on the quantification of semantic change rather than the underlying word sense induction because form-based approaches consistently outperformed sense-based approaches. However, the clustering algorithms that have been employed do not take the temporal nature of documents into consideration, and we thus argue that they are not optimal for modeling word meaning over time.

In this chapter, we have outlined several possible paths forward, both in terms of diachronic word sense induction and assessment of change. Each proposed path is suitable for different kinds of research questions and data. For example, by clustering embeddings over a whole corpus, smaller senses that would not appear in sequential modeling can gain sufficient evidence in global clustering. Such a method is however computationally expensive. Other methods suffer from the problem that when only consecutive time periods are considered, slow and gradual shift risks being missed and over long time periods other strategies are more suitable. Among these methods, we strongly advocate for a shift towards incremental methods as these are currently the best fit to the LSC problem.

Chapter 5

A novel, evolutionary clustering algorithm: A-Posteriori affinity Propagation

"Birds of a feather flock together"

English proverb

5.1 Introduction

In the previous chapter, we have outlined that the capability to perform text clustering by considering the temporal nature and progression of data is a crucial aspect for modeling lexical semantic change. Thus far, word meanings have been modeled through conventional clustering algorithms. Among these algorithms, Affinity Propagation has gotten more and more popular over standard algorithms like K-Means (Park et al., 2022; Martinc et al., 2020b; Alagic et al., 2018). However, Affinity Propagation (Frey and Dueck, 2007), as well as K-Means (MacQueen et al., 1967) and other conventional clustering algorithms, is mostly conceived to deal with static datasets, where all the objects are available as a whole and clustering is performed offline over the entire set of data (Sun and Guo, 2014). Extensions based on incremental solutions are proposed to deal with dynamic datasets, where objects continuously arrive, and clustering is performed by processing new data as they appear. Instead of recomputing the clustering from scratch every time new objects are received, *incremental clustering* algorithms aim to efficiently update the clustering by processing and assimilating the new objects into the existing clusters.

Scalability issues become relevant in designing incremental clustering algorithms for dynamic datasets, as they have to cope with high data volumes, sequential access, and the dynamically evolving nature of the data to be classified. To support temporal evolution analysis and to trace cluster changes over time, *evolutionary* incremental clustering algorithms have been proposed, generating a sequence of clustering results, one for each time period (Beringer and Hüllermeier, 2006; Hruschka et al., 2009). Two main issues become relevant in evolutionary clustering. A first issue regards the *faithfulness* property, that is, the clustering at any point in time should remain faithful to the current data as much as possible, thus avoiding resulting clusters to dramatically change from one time step to the next (Chakrabarti et al., 2006). This property facilitates the exploitation of clustering results over time, namely the capability to trace the *cluster history*, since users get progressively familiar with results and can compare clustering of different time periods in a more effective way. A second issue regards the so-called *stability-plasticity dilemma*, that is, the phenomenon by which "some patterns may be lost to learn new knowledge, and learning new patterns may overwrite previously acquired knowledge" (Yang et al., 2013). Thus, faithfulness is enforced in evolutionary clustering to learn new information without forgetting what has been previously learned. As an additional property, *forgetfulness* is required to discard information that become obsolete, thus reducing memory usage and enforcing scalability.

Chapter outline.

This chapter includes materials originally published in the following publication:

> Silvana Castano, Alfio Ferrara, Stefano Montanelli, and Francesco Periti. 2024. Incremental Affinity Propagation based on Cluster Consolidation and Stratification. eprint 2401.14439, arXiv. Under review.

In this chapter, we propose an incremental extension of the Affinity Propagation (AP) algorithm, which has been extensively used for LSC and various linguistic tasks such as word sense induction (Alagic et al., 2018; Kutuzov et al., 2017). Our extension is called *A-Posteriori affinity Propagation* (APP) and is based on *cluster consolidation* and *cluster stratification* to achieve faithfulness and forgetfulness. Although APP is designed for application in LSC, benchmarks with diachronic sense labels spanning multiple time periods do not currently exist at the time of this thesis. Thus, we decided to first evaluate its performance against benchmark algorithms in a standard clustering setting and then assess its applicability to LSC. This chapter will focus on the formal definition of APP and its evaluation against benchmark AP algorithms. We will address its applicability to LSC in the next chapter.

The chapter is organized as follows. In Section 5.2, the traditional AP algorithm as well as its main incremental extensions are over-viewed. We introduce the APP algorithm in Section 5.3. In Section 5.4 and 5.5, we present the evaluation setup and results of our evaluation, respectively. Finally, we provide a brief summary of this chapter in Section 5.6, and we refer to the next chapter for a thorough illustration and discussion about the applicability of APP to LSC.

5.2 Background and related work

Work related to incremental clustering over dynamic datasets and temporal/stream-based data aggregation techniques is widely discussed in the literature (e.g., Mansalis et al., 2018; Silva et al., 2013; Mei and Zhai, 2005). In this chapter, the APP algorithm we are proposing is conceived as an extension of the original AP algorithm (Frey and Dueck, 2007). For this reason, in the following, we first recall the main features of AP, and then we review the main incremental extensions of this algorithm, by also highlighting the distinctive features of our APP algorithm with respect to the considered solutions.

5.2.1 Affinity Propagation

Affinity Propagation (AP) is a clustering algorithm based on “message passing" between data points represented as connected nodes on a bipartite graph, in which edges represent the similarity between pairs of points. The main advantage is that, unlike other clustering algorithms such as K-Means or K-Medoids, it does not require the number of clusters to be determined beforehand since they are formed around exemplary nodes, namely *exemplars*, which are representative nodes of the clusters. The objective function is to maximize

$$z = \sum_{i=1}^{n} s(i, c_i) + \sum_{k=1}^{n} \delta_k(\mathbf{c}) \tag{5.1}$$

where $s(i, c_i)$ denotes similarity between a node $\mathbf{x}_i$ and its nearest exemplar $\mathbf{x}_{c_i}$, and $\delta_k(\mathbf{c})$ has the form

$$\delta_k(\mathbf{c}) = \begin{cases} -\infty & \text{if } c_k \neq k \text{ but } \exists i : c_i = k \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

and penalises invalid configurations where a node $\mathbf{x}_i$ chooses another nodes $\mathbf{x}_k$ as its exemplar without $\mathbf{x}_k$ being labelled as an exemplar. The optimization problem is implemented by exchanging two kinds of messages between nodes on the graph:

1. *responsibility* $r(i, k)$, sent from node $\mathbf{x}_i$ to the candidate exemplar $\mathbf{x}_k$ indicates to what extent $\mathbf{x}_k$ is a good exemplar for $\mathbf{x}_i$.

2. *availability* $a(i, k)$, sent from the candidate exemplar $\mathbf{x}_k$ to node $\mathbf{x}_i$ indicates to what extent it would be for $\mathbf{x}_i$ to choose $\mathbf{x}_k$ as its exemplar taking into account the accumulated evidence obtained from other nodes about the suitability of $\mathbf{x}_k$ as an exemplar.

According to Frey and Dueck (2007), $r(i, k)$ and $a(i, k)$ can be computed as follows:

$$r(i, k) \leftarrow s(i, k) - \max_{k',\, k' \neq k} \left\{ a(i, k') + s(i, k') \right\} \tag{5.3}$$

$$a(i, k) \leftarrow \min \left\{ 0, r(k, k) + \sum_{i',\, i' \notin \{i, k\}} \max \left\{ 0, r(i', k) \right\} \right\} \tag{5.4}$$

Unlike the other pairs, the so called *self-availability* $a(k, k)$ is computed as

$$a(k, k) = \sum_{i', i' \neq k} \max \left\{ 0, r(i', k) \right\} . \tag{5.5}$$

In the beginning, all messages are initialized to 0. Then, AP iteratively updates responsibilities and availabilities until convergence. The number of resulting clusters is determined by the clustering algorithm. However, it was argued by Frey and Dueck (2007) that it is influenced by the self-similarity value $s(i, i)$, which is called *preference*, and by the *damping* factor which damps the responsibility and availability of messages to avoid numerical oscillations in the updates.

As a general remark, Frey and Dueck (2007) suggest preference p should be the median, or minimum value of similarities and point out that a larger p generates a larger number of clusters. The damping factor d should be at least 0.5 and less than 1. In particular, the responsibility and availability messages are "damped" as follows

$$\mathbf{msg}_{new} = d \cdot \mathbf{msg}_{old} + (1 - d) \cdot \mathbf{msg}_{new} \tag{5.6}$$

where $\mathbf{msg}_{old}$ and $\mathbf{msg}_{new}$ are the values of $a(i, k)$ and $r(i, k)$ before and after the update, respectively.

5.2.2 Incremental extensions of Affinity Propagation

AP was designed for discovering patterns in static data. Several extensions have been proposed to cope with data appearing in a dynamic manner. Incremental extensions of AP have been successfully employed in a series of problems such as text clustering (Shi et al., 2009), robot navigation (Ott and Ramos, 2012), and multi-spectral images classification (Yang et al., 2013). Moreover, we also consider incremental AP extensions where a notion of *clustering history* is somehow supported, that is the capability to trace the object membership over time or to compare clusters related to different time steps. A comparative overview of the considered AP extensions is provided in Table 5.1.

STRAP: Streaming AP. Zhang et al. (2008) propose an incremental AP clustering algorithm (STRAP) for data streaming settings that reduces the time complexity of AP by limiting the number of its re-computations. The idea is to assign new objects to previously generated clusters only if they satisfy a similarity requirement with respect to the current exemplars. On the contrary, a reservoir is leveraged to detain too dissimilar objects. When the size of the reservoir exceeds a threshold, or some changes in the rate of acquisition are detected, the AP is re-executed over the current exemplars and the objects in the reservoir. An additional step is employed to merge the exemplars independently learned from subsets of the whole dataset.

I-APC: Incremental AP clustering. Shi et al. (2009) propose a semi-supervised incremental AP (I-APC) which injects some supervision in the clustering by adjusting the similarity matrix of the AP algorithm. They set much larger distances for objects with the same label and much smaller distances for objects with different labels. At each time step, after each AP run, the labeled dataset is extended with the most similar objects to the current clusters, and the similarity matrix is reset according to the newly labeled data. However, this step affects computational time and it makes I-APC cost more CPU time than AP.

ID-AP: Incremental and Decremental Affinity Propagation. Similarly to Shi et al. (2009), Yang et al. (2013) propose a semi-supervised incremental algorithm, called Incremental and Decremental AP (ID-AP), that in-

Work	Learning	Basic algorithms	Clustering history	Efficiency	Description
STRAP (Zhang et al., 2008)	Unsupervised	AP	No	faster than AP	STRAP assigns new objects to previously generated clusters based on their similarity.
I-APC (Shi et al., 2009)	Semi-supervised	AP	No	slower than AP	I-APC injects supervision in AP by adjusting the similarity matrix.
ID-AP (Shi et al., 2009)	Semi-supervised	AP	No	slower than AP	ID-AP injects supervision in AP by adjusting the similarity matrix, and discard useless labeled objects at each time step.
IAPKM (Sun and Guo, 2014)	Unsupervised	AP K-Medoids	No	faster than AP	IAPKM adjusts the current clustering results according to new objects by combining AP and K-Medoids.
IAPNA (Sun and Guo, 2014)	Unsupervised	AP Nearest Neighbors	No	faster than AP	In IAPNA, responsibilities and availabilities of the new objects are assigned referring to their Nearest Neighbor among the previous objects.
EAP (Arzeno and Vikalo, 2021, 2017)	Unsupervised	AP	Yes	faster than AP	EAP trace the clustering history by introducing consensus nodes and factors into the AP graph.
SED Stream-AP (Sunmood et al., 2018)	Unsupervised	AP SED-Stream	Yes	slower than AP	SED Stream-AP trace the clustering history by combining the SED-Stream and AP clustering algorithms.
APP (Castano et al., 2024) (Periti et al., 2024e, 2022)	Unsupervised	AP	Yes	faster than AP	APP traces the cluster history by consolidating past clustering results through the use of cluster centroids, and discards obsolete objects at each time step by enforcing cluster pruning.

Table 5.1: Summary view of incremental extensions of AP.

corporates a small number of labeled samples to guide the clustering process of the conventional AP algorithm. At each time step the labeled samples are used as prior information to adjust the similarity matrix of the AP algorithm. Furthermore, the algorithm deals with the *stability-plasticity* dilemma by using an incremental and a decremental learning approach for selecting the most informative unlabeled data and discarding useless labeled samples, respectively. The intrinsic relationship between the labeled samples and unlabeled data improves the clustering performance. On the other hand, the learning phase of ID-AP method is several times higher than that required from the conventional AP since the selection/discard phases involve repeated execution of the clustering algorithm.

IAPKM: Incremental Affinity Propagation based on K-Medoids. Sun and Guo (2014) present an Incremental Affinity Propagation based on K-Medoids (IAPKM). The goal of this extension is to adjust the current clustering results according to new incoming objects, rather than recomputing AP clustering on the whole data set. IAPKM combines AP and K-Medoids in an incremental clustering task, that is: AP clustering is executed on the initial bunch of objects, and K-Medoids is employed to modify the current clustering result according to the new arriving objects. As a result, IAPKM achieves comparable clustering performance and can save a great deal of time compared to the conventional AP algorithm. However, the number of clusters cannot be adjusted according to the new incoming objects since the traditional K-Medoids can't adjust the number of clusters automatically.

IAPNA: Incremental Affinity Propagation based on Nearest Neighbor Assignment. As an alternative to IAP-KM, Sun and Guo (2014) discuss an Incremental version of Affinity Propagation based on Nearest Neighbor Assignment (IAPNA). The intuition under IAPNA is that objects added at different time steps are at different statuses: pre-existing objects have established certain relationships (nonzero responsibilities and nonzero availabilities) between each other after AP, while new objects' relationships with other objects are still at the initial level (zero responsibilities and zero availabilities). The idea of IAPNA is to put all the data points at the same status before proceeding with the AP procedure till convergence. According to this idea, responsibilities and availabilities of the new incoming objects are assigned referring to their nearest neighbors. Similarly to IAPKM, IAPNA achieves higher performance than traditional AP clustering while reducing computational complexity. In addition, it preserves the AP feature of automatically discovering new clusters.

EAP: Evolutionary Affinity Propagation. An Evolutionary Affinity Propagation (EAP) is presented by Arzeno and Vikalo; Arzeno and Vikalo (2021; 2017). Compared to previous incremental extensions of AP, EAP is the first algorithm that can automatically trace the clustering history and temporal changes in cluster memberships across time. EAP introduces consensus nodes and factors into the AP graph with the aim to encourage objects to select a consensus node, rather than another object, as their exemplar. Clusters are traced by observing the positions of consensus nodes in the clustering history. Basically, the creation and the disappearance of consensus nodes indicate cluster birth and death, respectively. In EAP, the computational time is also reduced since messages need to be passed between consensus nodes and not between all pairs of objects.

SED Stream-AP: Evolutionary Affinity Propagation. Sunmood et al. (2018) propose the evolutionary clustering SED-Stream-AP as an integration of the SED-Stream (Waiyamai et al., 2014) and the AP clustering algorithms. SED-Stream-AP adopts a two-stage process phases, called *online* and *offline* phase, respectively. In the online phase, the clustering history is continuously monitored and detected. The evolution-based clustering of SED-Stream enables SED-Stream-AP to support different evolving structures (e.g., appearance, merge). In the offline phase, the AP clustering is used to automatically determine the number of clusters and deliver the final clustering without any need for user intervention.

5.2.3 Framing APP with respect to the existing solutions

- Inspired by STRAP, APP performs clustering over exemplars created in past aggregation stages and new incoming objects. In contrast to STRAP, new incoming objects are *a posteriori* clustered and not *a priori* assigned to a previously generated cluster. In addition, APP replaces the use of a reservoir with the assumption of "group evolution", meaning that a new cluster for a new kind of objects can be detected only if there is a relevant number of incoming exemplar objects associated with it.

- In contrast to I-APC, APP is completely unsupervised and does not inject supervision in the similarity between objects.

- Similarly to ID-AP, APP is an incremental extension of AP conceived for dealing with the stability-plasticity dilemma by enforcing *faithfulness* and *forgetfulness* in evolutionary scenarios.

- In contrast to IAPKM, APP relies entirely on the AP algorithm, enabling the number of clusters to be adjusted automatically.

- Similar to IAPNA, APP considers the relationships established by pre-existing objects. However, while IAPNA considers all these relationships individually, APP consolidates them into cluster exemplars, which will represent the entire clusters in the following iterations.

- Like EAP and SED-Stream-AP, APP can trace the clustering history by supporting different kinds of cluster stratifications.

Specifically, APP enforces *incremental* clustering where i) new arriving objects at time t are dynamically consolidated into previous clusters at time $t - 1$ without the need to re-execute clustering over the entire dataset of objects, and ii) a faithful sequence of clustering results is produced and maintained over time, while allowing to forget obsolete clusters with *decremental* learning functionalities. Cluster consolidation means that APP keeps the memory of clustering results at time $t-1$ by collapsing each cluster into a summary representation, namely the *centroid*, which is considered as an additional object to cluster at time t. Cluster stratification means that the new clusters at time t are obtained from clusters at time $t-1$ by i) creating a new cluster including new objects arriving at time t (*stratification-by-creation*), ii) inserting new objects arriving at time t into an existing $t - 1$ cluster (*stratification-by-enrichment*), iii) merging two or more $t - 1$ clusters into a new one at time t (*stratification-by-merge*).

APP can be used for discovering concepts in incremental scenarios under the assumption of "**group evolution**", in contrast to the "individual evolution". A new incoming object dissimilar from the past observations tends to be considered by APP as an outlier of a previously generated cluster rather than a unique exemplar of a new cluster. This means that a new cluster can be detected only if there is a relevant number of incoming exemplars associated with it. Finally, to enforce forgetfulness, a decremental learning functionality is defined in APP to allow the selective pruning of aged, obsolete clusters, similarly to the *forgetful property of human mind* (Yang et al., 2013).

5.3 A-Posteriori affinity Propagation

Using the conventional AP algorithm to cluster dynamic datasets is not suitable to cope with the stability-plasticity dilemma (Yang et al., 2013). In particular, clusters generated at time $t-1$ can be mixed up due to a new bunch of objects that arrive at time t (see Chapter 4). This means that previously clustered objects at time $t-1$ can remain in the same cluster at time t, but they can also be moved to another cluster due to the updated object picture from time $t-1$ to time t. In this situation, tracing the history of a specific cluster across different time periods becomes arduous, and a number of noisy clusters could be created when different kinds of objects arrive according to a skewed distribution (Martinc et al., 2020b).

Figure 5.1 shows an example of AP clustering illustrating such a problem. The conventional AP clustering is implemented on the initial bunch of objects ($t = 0$), represented by white circles. The clustering result is shown in Figure 5.1 (A), where the black objects denote the cluster exemplars. The new objects represented by gray diamonds and triangles arrive at time $t = 1$ and $t = 2$, respectively. After the arrival of new objects, the clustering result of the second and third AP run is shown in Figure 5.1 (B-C). By comparing Figure 5.1 (A-B-C), we note that some objects change cluster in the various AP rounds and several clusters are generated ($t = 2$).

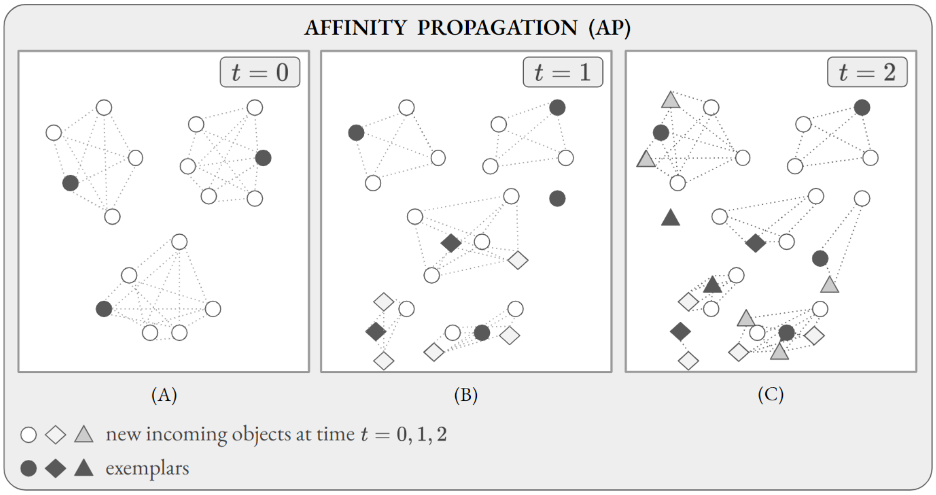

AFFINITY PROPAGATION (AP)
$t = 0$
$t = 1$
$t = 2$
(A)
(B)
(C)
○ ◇ △ new incoming objects at time $t = 0, 1, 2$
● ◆ ▲ exemplars

Figure 5.1: Example of AP with an incremental scenario. (A) shows the clustering result over the initial bunch of objects ($t = 0$) represented by white circles. The black objects denote the cluster exemplars and dashed lines connect the objects of each cluster. (B) show the clustering result after the second AP run ($t = 1$). New incoming objects at time $t = 1$ are represented by gray diamonds. Similarly to (B), the clustering result after the third AP run ($t = 2$) is shown in (C). New incoming objects at time $t = 2$ are represented by gray triangles.

In the following, we present APP. The objects to cluster become progressively available at different time steps $t = \{0, \ldots, n\}$. At each time step t, APP clusters the new incoming objects *a-posteriori* by considering a consolidated version of the clusters created at time $t-1$. For each cluster, the AP notion of exemplar is replaced by *centroid* and it is defined as a summary representation of the associated objects with the aim to consolidate the cluster observed until $t - 1$. In particular, we work with objects that are data points, namely vectors of numerical features. In this context, a cluster centroid is computed as an average representation of the associated object vectors. As a main difference with AP, in APP, the objects previously clustered do not change cluster when new objects arrive and clusters generated in a certain time step are consolidated/stratified over the past ones.

5.3.1 The APP algorithm

Algorithm 1 provides the pseudo-code of the proposed APP.

Algorithm 1 *The APP algorithm*

```
Input
$t$: time step
$X$: objects at time step t
$X_1$: objects at time step $t - 1$
$L_1$: labels at time step $t - 1$
$th_\gamma$: pruning threshold

Output
$L, X$: at time step t

if t == 0 then
    L ← $AP$(X)

else
    $\mu X_1 \leftarrow Pack(L_1, X_1)$
    $L_2 \leftarrow AP(\mu X_1 \cup X)$
    $\mu L_1, L \leftarrow Split(L_2)$
    $L_1 \leftarrow UnpackAndUpdate(\mu L_1, \mu X_1, L_1, X_1)$
    $L, X \leftarrow Pruning(L_1 \cup L, X_1 \cup X, th_\gamma)$
end if

yield L, X
```

Let's call X and X_1, and L and L_1 the objects and the cluster labels at time t and $t - 1$, respectively. At time $t = 0$, the execution of APP coincides with the conventional AP algorithm. At each time $t > 0$, for each existing cluster computed at time $t - 1$, the objects $x_i \in X_1$ are packed into a single representation called cluster centroid μ. The set of the centroids for X_1 is denoted μX_1. Then, the conventional AP algorithm is executed on $\mu X_1 \cup X$, with the aim to obtain a new set of temporary labels L_2, i.e., the new assignment of objects to clusters. Such labels are then split into two subsets, μL_1 and L, which contain labels for each average representation in μX_1 and for each object in X, respectively. Given $\mu L_1, \mu X_1, L_1, X_1$, APP unpacks the centroids of μL_1 into the corresponding objects X_1 mapping the previous labels L_1 into the new labels of their respective centroids μL_1. Finally, APP returns $L_1 \cup L$, which is the union of the unpacked and updated L_1 and L.

The APP algorithm enforces *faithfulness* and *forgetfulness* as described in the following.

Faithfulness is the capability to preserve clustering history possibly enriched with new objects. At time t, the execution of APP ensures that the objects X_1 arrived in previous time steps do not change cluster. Indeed, each cluster existing at time $t-1$ is summarised by a centroid defined as an average representation of the cluster objects associated with it through L_1. The centroids are not changed by the APP execution at time t, thus also the objects arrived until $t-1$ cannot change cluster. As a result, the clusters of time $t-1$ and the associated centroids constitute the "memory" of the objects observed in the past. In APP, the centroids of clusters at time $t-1$ are exploited as additional objects to cluster together with the new incoming objects at time $t > 0$. The new objects are stratified over the existing clusters according to one of the following criteria:

- *stratification-by-creation*: a new cluster is created containing a subset of the new incoming objects $\bar{X} \subseteq X$ when all the objects in $\bar{X}$ are found to be too dissimilar from all the existing cluster centroids μX_1.
- *stratification-by-enrichment*: a previously created cluster is enriched with a subset of the new incoming objects $\bar{X} \subseteq X$ when all the objects in $\bar{X}$ are found to be similar to a cluster centroid in μX_1.
- *stratification-by-merge*: a new, unique cluster is created by merging two or more centroids in μX_1 and a subset of the new incoming objects $\bar{X} \subseteq X$ when the objects in $\bar{X}$ are found to be similar to all the merged centroids.

Forgetfulness is the capability to recognize obsolete clusters and discard them. At a certain time t, it is possible that a cluster represents the memory of a group of *obsolete objects*, namely a group emerged in past time steps, but disappeared in recent observations. To enforce forgetfulness, APP allows to drop the clusters that represent obsolete groups of objects. Each cluster is associated with an *aging index* $\gamma \leq t$ that denotes the last time step t in which the cluster has been created/changed. For instance, a cluster enriched by new objects at time t has an aging index $\gamma = t$. A *pruning threshold* $th_\gamma \in [1, +\infty]$ is defined in APP to define when a cluster can be considered obsolete. The threshold specifies the maximum number of APP rounds that can be executed without any change on a cluster contents. At each time step, each cluster defined by L is evaluated for possible pruning with respect to th_γ. Given a cluster with aging index γ, the cluster is pruned when $t - \gamma > th_\gamma$. When $th_\gamma \geq t$, it means that forgetfulness is not enforced and all the clusters created at any time step is maintained. Otherwise, forgetfulness is enforced and the pruning condition is applied. For instance when $th_\gamma = 1$ and $th_\gamma < t$, all the clusters not enriched at the last time t are considered obsolete, and then pruned.

Figure 5.2 is an example of APP execution with pruning threshold $th_\gamma = 1$. The initial bunch of objects ($t = 0$) is shown in Figure 5.2 (A). The clustering result at time $t = 0$ is represented in Figure 5.2 (B). Black objects denote the cluster exemplars. In Figure 5.2 (C), centroids are calculated as average representations of cluster objects ($t = 1$) and they are denoted as bold circles. New objects at time ($t = 1$) are represented as gray diamonds in Figure 5.2 (D). After the cluster consolidation, the clustering result of the APP run is shown in Figure 5.2 (E) ($t = 1$). In particular, Figure 5.2 (E) shows an example of stratification-by-creation (i.e., cluster on the bottom-left corner) and an example of stratification-by-enrichment (i.e., cluster on the bottom-middle part). In Figure 5.2 (F), each centroid is unpacked and its cluster label is associated to each object it had previously packed. The consecutive round of APP ($t = 2$) is presented in Figure 5.2 (G-H-J). In particular, Figure 5.2 (I) shows an example of stratification-by-merge where two previously generated clusters are merged into a single one. The final clustering result at time $t = 2$ is shown in Figure 5.2 (J). As a result of the stratification-by-pruning, the cluster on the right-top corner in Figure 5.2 (I) is pruned in Figure 5.2 (J) since it is unchanged for two iterations. As a difference with AP (see Figure 5.1), objects do not change cluster in Figure 5.2 and a lower number of clusters is generated.

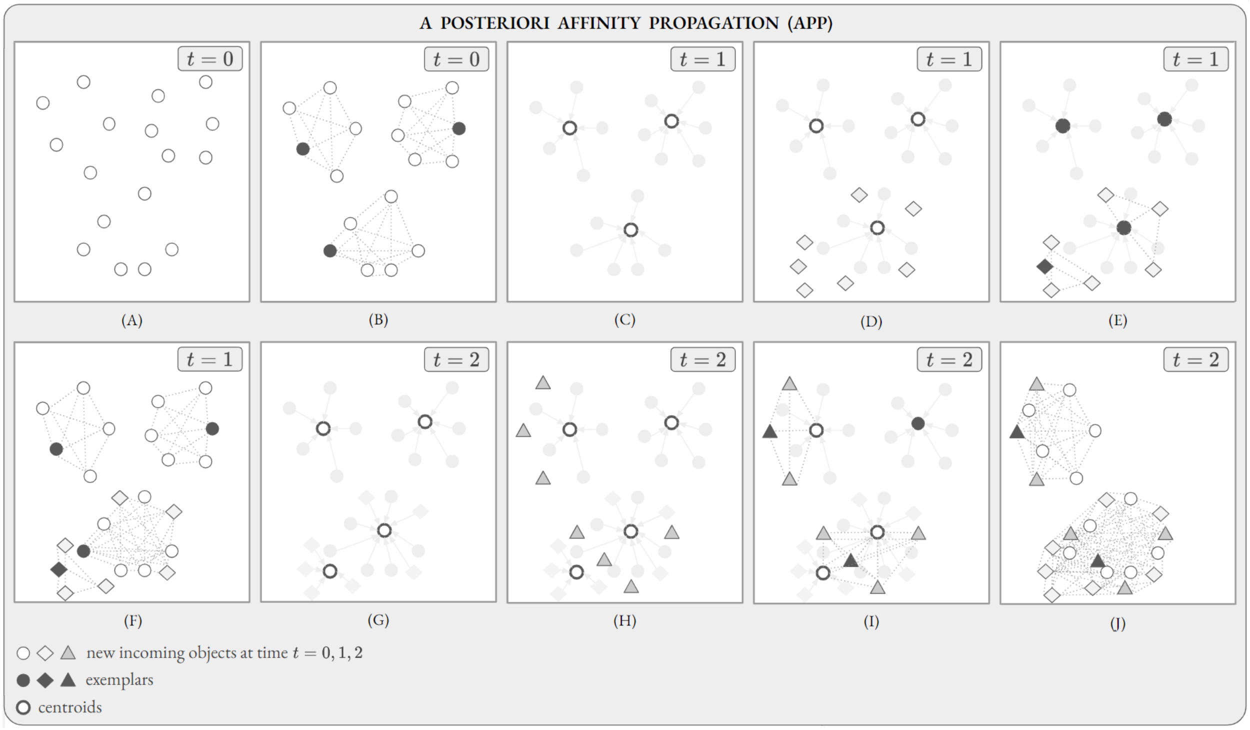


Figure 5.2: Example of APP. (A) shows the objects available at time $t = 0$. The first clustering result coincides with AP and it is represented in (B). The black objects denote the cluster exemplars. For the sake of clarity, dashed lines fully connect the objects of each cluster. (C) shows the cluster centroids as bold circles generated by averaging the objects of each cluster on the background. (D) shows the input objects of APP at time $t = 1$. Gray diamonds represent the new incoming objects. The clustering result is represented in (E). In (F), cluster centroids are unpacked and their cluster labels are associated with each object they previously packed. The second APP run at time $t = 2$ is shown in (G)-(H)-(J). New incoming objects are represented by gray triangles. (J) denotes the final clustering result. Note that the cluster on the right-top corner of (I) disappears in (J) due to a pruning threshold $th_\gamma = 1$.

5.3.2 Complexity and Memory Usage Analysis

Since APP leverages AP for object clustering, the complexity of APP and AP are related. In AP, the time complexity of message-passing iteration according to Equations 5.3 and 5.4 is $\mathcal{O}(N^2)$, where N is the number of all the current available objects. Therefore, the time complexity is $\mathcal{O}(N^2T)$, where T is the number of iterations until convergence. Further, the memory complexity is in the order $\mathcal{O}(N^2)$ if a dense similarity matrix is used.

Similarly, the time complexity of APP is $\mathcal{O}(M^2T_1)$, where $M = (\mu_{t-1} + n_t)$, and μ_{t-1}, n_t are the number of previous centroids and the number of the new incoming objects, respectively. At each iteration, the memory complexity of APP is $\mathcal{O}(M^2)$, in that, there is no need to keep in memory previously clustered objects during the AP execution of APP (Algorithm 1, row 6). By definition $M << N$ and $T_1 << T$, thus a lot of time and memory is saved, making APP a scalable solution in incremental scenarios. Moreover, when $th_\gamma > 0$, time and memory complexity are further reduced to $\mathcal{O}(M_\gamma^2 T_2)$, $\mathcal{O}(N_\gamma)$, respectively; where $M\gamma = (\mu_{t-1}^{(\gamma)} + n_t)$ and $\mu_{t-1}^{(\gamma)}$ is the number of previous centroids that were not affected by pruning, and $T_2 < T_1$. Basically, the smaller γ, the more $\mu_{t-1}^{(\gamma)} < \mu_{t-1}$, since more clusters will be pruned.

5.4 Experimental setup

The goal of our experimentation is to compare the results of APP against benchmark clustering algorithms. We note that official implementations of incremental AP algorithms are not available for comparison. We thus selected AP since it is the baseline clustering algorithm on which APP relies upon, and IAPNA since it is a well-known and top-cited incremental extension of AP, being also straightforward to implement at the same time. In the evaluation, we first focus on two evaluation experiments called **uniform-incremental** and **variable-incremental** experiments. Both the experiments are based on a dynamic scenario where the objects to cluster arrive as separated bunches at different time steps. In the uniform-incremental experiment, we define the number and the set of objects arriving at the various time steps without any constraint on the category. The idea is to analyze the behavior of the considered clustering algorithms on a pure incremental setting like the one proposed in Sun and Guo (2014) (see Section 5.4.1). In the variable-incremental experiment, the category of the objects arriving at each time step is constrained according to a given schema. The idea is to analyze the capability of the considered clustering algorithms to recognize the categories of the incoming objects when they appear over time according to a specific incremental schema, that can be growing, shrinking, or stable (see Section 5.4.1).

All the experiments are implemented in Python 3.10 and they are conducted on a PC with 1.80GHz Intel Core i7 processor and 16GB of RAM. Our code is based on the implementation of AP by scikit-learn (Pedregosa et al., 2011).[1] The APP code is available at `https://github.com/umilISLab/APP`.

[1] `scikit-learn.org/stable/`

Datasets and pre-processing. In the evaluation, four popular labeled datasets are considered. In particular, we selected Iris, Wine, and Car datasets from Newman et al. (1998) since they are used in the evaluation of AP and IAPNA by Sun and Guo (2014). Moreover, we added the KDD-CUP dataset since it is characterized by a high number of categories (Sunmood et al., 2018), and thus it is appropriate for clustering evaluation in incremental experiments. In all the datasets, the objects are described as feature vectors; a different number of features per object is defined for each dataset.

A summary view of the benchmark datasets used in the evaluation is provided in Table 5.2.

Dataset	Number of objects	Number of features	Number of categories	Usage of dataset
Iris	150	4	3	whole
Wine	178	13	3	whole
Car	260	6	4	partly
KDD-CUP	2904	41	11	partly

Table 5.2: A summary description of the benchmark datasets.

Some datasets (Car and KDD-CUP) are characterized by a highly unbalanced number of objects per category. As in Sun and Guo (2014), we select and use only part of them. In particular, we consider 65 objects taken from the top 4 most numerous categories in the Car dataset, and 264 objects taken from the top 11 most numerous categories in the KDD-CUP dataset.

A pre-processing stage is enforced to normalize the dataset objects. Since the experiments are performed in a dynamic scenario, a single normalization stage on the whole dataset is not appropriate. Instead, at each time step of the experiments, we perform normalization on the N_t objects of the dataset available at time t . For the sake of comparison, we use the same normalization used by Sun and Guo (2014).

Evaluation metrics. As in Sun and Guo (2014), for clustering objects, we calculate the similarity between pairs of objects through the negative euclidean distance where we do not leverage the preference coefficients described by Sun and Guo (2014). For each dataset, the preference p (self-similarity) is set to the median of the input similarities at a given time (see Section 5.2 for further details about the p parameter).

The clustering results are evaluated according to *Purity* (PUR) and *Normalized Mutual Information* (NMI). To compute PUR, each cluster is assigned to the category that is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned objects and by dividing by N_t, that is the number of objects of the dataset available at time t. Formally:

$$PUR(\Omega, C) = \frac{1}{N_t} \sum_k \max_j \ \bar{\omega}_k \cap \bar{c}_j \ , \tag{5.7}$$

where $\Omega = \{\omega_1, ..., \omega_K\}$ is the set of clusters, $C = \{c, ..., c_J\}$ is the set of categories, and $\bar{\omega}_k$ and $\bar{c}_j$ are the set of objects in ω_k and c_j, respectively. High PUR values are frequently achieved when a high number of clusters is generated. For instance, PUR is 1 when each object is placed in a corresponding singleton cluster. Thus, we also exploit NMI to estimate the quality of the clustering by considering the number of generated clusters. NMI is defined as:

$$NMI(\Omega, C) = \frac{I(\Omega, C)}{[H(\Omega) + H(C)]/2}, \tag{5.8}$$

where $I(\Omega, C)$ is the mutual information between the set of clusters Ω and the set of categories C, and the normalization $[H(\Omega) + H(C)]/2$ is introduced to penalise large cardinalities of Ω with respect to C, in that, the entropy $H(\Omega)$ tends to increase with the number of clusters.

As in Sun and Guo (2014), three metrics are employed to evaluate the scalability of the considered clustering algorithms, namely the *Number of Iterations* until convergence (NI), the *Computation Time* (CT) in seconds, and the *Memory Usage* (MU) in MB. Furthermore, we also consider the *Number of Clusters* (NC) generated at each time step.

5.4.1 Experimental settings

The *uniform-incremental* and *variable-incremental* settings are discussed in the following.

As a general remark, we stress that the experiments are repeated 100 times for each dataset; each time, the order of incoming objects is randomly defined. For each dataset, the settings of the 100 executions are stored and used for each considered algorithm (i.e., AP, IAPNA, and APP). We analyze the results by considering the median score of the 100 obtained values at each time step.

The hyper-parameters of the AP algorithm are configured as follows: the maximum number of iterations is set to 200, the damping factor is set to 0.9, and 15 iterations without changes in the exemplars at the last time step are required before declaring convergence.

About IAPNA, since the implementation used in the evaluation of Sun and Guo (2014) is not available, we developed a Python IAPNA implementation for the sake of our experiments.

About the APP configuration, we define a pruning threshold $th_\gamma = 1$.[2]

Uniform-incremental setting

In the uniform-incremental setting, we borrow the evaluation setup proposed by Sun and Guo (2014). A fixed (i.e., uniform) number of objects is scheduled for arrival at any time step without considering the category. Each dataset is shuffled and split through sampling into six bunches (one for each time step). For each dataset, we define i) the number of incoming objects at the first time step ($t = 0$), and ii) the number of incoming objects at any subsequent time steps ($t > 0$). In this experiment, most of the objects become available at time step 0-th, while few objects are introduced in the subsequent time steps. The details about dataset sampling in the incremental setting are provided in Table 5.3. For instance, considering the IRIS dataset, 100 objects are sampled for clustering at the first time step, and 10 by 10 objects are sampled in the subsequent time steps.

[2] As pruning threshold, we chose the value that provided the best trade-off between APP performance and scalability in all the considered experiments.

Dataset	Number of objects (first time step) objects	Number of objects (subsequent time steps)
Iris	100	10
Wine	128	10
Car	210	10
KDD-CUP	1904	200

Table 5.3: The number of objects in the uniform-incremental setting (first and subsequent time steps).

Variable-incremental setting

In the variable-incremental experiment, the number of incoming objects at each time step is not fixed/uniform. The goal is to analyze the behavior of clustering algorithms when a larger number of incoming objects is scheduled for arrival at each time step with respect to the uniform-incremental experiment. Moreover, the category of the objects arriving at each time step is chosen according to a specific incremental schema. Each dataset is shuffled and split through sampling into six bunches (one for each time step). The object sampling from each category in a given time step is defined according to one of the following schema/behavior:

1. *growing*, the objects of a category are sampled by scheduling the order of arrival to be ascending in size across the time steps. The category reproduces the behavior of a growing group of objects over time.

2. *shrinking*, the objects of a category are sampled by scheduling the order of arrival to be decreasing in size across the time steps. The category reproduces the behavior of a shrinking group of objects over time.

3. *stable*, an equal number of objects of a category is scheduled for arrival in any time step. The category reproduces the behavior of a stable group of objects over time.

In each of the 100 iterations, each category of the datasets is associated with a certain schema with a 33% probability (i.e., the three schemas are equally probable over the categories). The arrival of objects of growing and shrinking categories can be focused in a subset of the time steps. This means that the objects of a growing category can start to appear in a time step $t > 0$, as well as the objects of a shrinking category can be consumed before the last time step. As a consequence, in a given time step, the objects of a category can be missing. Otherwise, according to the "group evolution" assumption, a minimum number of objects q of a category is scheduled for arrival in any time step t according to the associated schema. The aim is that any category appearing in a certain time step has enough objects for being recognized by the clustering algorithms. As a final constraint, we define that the incoming objects at each time step are taken from two different categories as a minimum.

In the experiment, for each category, we define q as the 10% of the dataset size divided by the number of dataset categories. A summary of q values for the categories of each dataset is provided in Table 5.4.

Dataset	*q* **parameter**
Iris	5
Wine	6
Car	7
KDD-CUP	26

Table 5.4: The minimum number of objects q per dataset category in the variable-incremental setting.

5.5 Experimental results

All the considered algorithms (i.e., AP, IAPNA, and APP) are based on AP for clustering objects in the first time step. Thus, the results of the three algorithms coincide with the first clustering execution at time $t = 0$. For this reason, the results on the 0-th bunch of objects are not shown/considered in the analysis.

Results on the uniform-incremental experiment

Experimental results with the uniform-incremental settings are shown in Tables 5.5, 5.6, 5.7, 5.8, 5.9, 5.10.

Dataset	**Method**	**1th**	**2th**	**3th**	**4th**	**5th**
Iris	AP	0.964*	0.975*	0.954*	0.957*	0.967*
	IAPNA	0.882	0.950	0.877	0.957*	0.953
	APP	**0.873**	**0.867**	**0.862**	**0.864**	**0.667**
Wine	AP	0.754	0.750*	0.747*	0.732*	0.730*
	IAPNA	0.884*	0.365	0.620	0.613	0.624
	APP	**0.710**	**0.655**	**0.665**	**0.661**	**0.663**
Car	AP	0.814*	0.830*	0.812*	0.816	0.812
	IAPNA	0.791	0.796	0.804	0.828*	0.823*
	APP	**0.727**	**0.604**	**0.704**	**0.514**	**0.550**
KDD-CUP	AP	0.863	0.812*	0.853	0.858	0.862
	IAPNA	0.349	0.515	0.512	0.983*	0.981*
	APP	**0.816**	**0.806**	**0.780**	**0.741**	**0.748**

Table 5.5: Uniform-incremental experiment: comparison on Purity. The highest score is denoted with an asterisk; the APP score is denoted in bold.

The results show that APP achieves comparable/higher clustering performance than the conventional AP and IAPNA algorithms. On average by considering all the time steps and datasets, APP achieves a PUR score of 0.724, which is comparable but lower than the PUR score of AP (0.846) and IAPNA (0.755). This result can be explained by considering the number of clusters NC created by the three algorithms, where we note that APP always returns the lowest value (see Table 5.10). As a matter of fact, a high number of clusters positively affects the PUR metric without considering the possible noisiness of the created groups. On the opposite, APP achieves a higher NMI score compared to AP and IAPNA. On average, APP obtains a NMI score of 0.553, while AP and IAPNA obtain 0.511 and 0.536, respectively. By considering the Wine and the Car datasets, we note that the NMI score of all three algorithms is quite low. This is probably due

Dataset	**Method**	**1th**	**2th**	**3th**	**4th**	**5th**
Iris	AP	0.600	0.660	0.586	0.561	0.568
	IAPNA	0.616	0.658	0.658	0.648	0.594
	APP	**0.707***	**0.740***	**0.712***	**0.718***	**0.734***
Wine	AP	0.346	0.339	0.335	0.329	0.326
	IAPNA	0.582*	0.000	0.484*	0.489*	0.565*
	APP	**0.363**	**0.444***	**0.444**	**0.445**	**0.417**
Car	AP	0.427	0.432*	0.417*	0.403	0.392
	IAPNA	0.415	0.409	0.403	0.406*	0.406*
	APP	**0.466***	**0.391**	**0.221**	**0.236**	**0.362**
KDD-CUP	AP	0.713	0.700	0.696	0.693	0.692
	IAPNA	0.564	0.668	0.665	0.754*	0.743*
	APP	**0.739***	**0.743***	**0.738***	**0.719**	**0.714**

Table 5.6: Uniform-incremental experiment: comparison on Normalized Mutual Information. The highest score is denoted with an asterisk; the APP score is denoted in bold.

Dataset	Method	1th	2th	3th	4th	5th
Iris	AP	0.128	0.117	0.319	0.321	0.156
	IAPNA	0.241	0.221	0.131	0.260	0.238
	APP	**0.009***	**0.008***	**0.010***	**0.009***	**0.008***
Wine	AP	0.199	0.182	0.204	0.221	0.278
	IAPNA	0.184	0.123	0.117	0.153	0.364
	APP	**0.052***	**0.047***	**0.051***	**0.050***	**0.051***
Car	AP	0.332	0.406	0.563	0.842	0.867
	IAPNA	0.200	0.678	0.282	0.844	0.231
	APP	**0.074***	**0.058***	**0.028***	**0.048***	**0.035***
KDD-CUP	AP	18.523	26.752	34.037	42.068	46.151
	IAPNA	44.656	43.041	36.304	83.318	68.759
	APP	**0.294***	**0.210***	**0.209***	**0.211***	**0.192***

Table 5.7: Uniform-incremental experiment: comparison on Computation Time. The highest score is denoted with an asterisk; the APP score is denoted in bold.

Dataset	**Method**	**1th**	**2th**	**3th**	**4th**	**5th**
Iris	AP	0.303	0.359	0.420	0.486	0.556
	IAPNA	0.308	0.366	0.428	0.496	0.569
	APP	**0.020***	**0.023***	**0.024***	**0.026***	**0.028***
Wine	AP	0.492	0.563	0.639	0.719	0.804
	IAPNA	0.507	0.581	0.659	0.742	0.831
	APP	**0.046***	**0.059***	**0.062***	**0.066***	**0.070***
Car	AP	1.215	1.325	1.440	1.559	1.684
	IAPNA	1.227	1.340	1.458	1.581	1.709
	APP	**0.050***	**0.055***	**0.058***	**0.037***	**0.034***
KDD-CUP	AP	108.287	129.658	153.012	178.233	205.425
	IAPNA	108.928	130.381	153.819	179.128	206.408
	APP	**2.207***	**2.850***	**3.029***	**3.207***	**3.400***

Table 5.8: Uniform-incremental experiment: comparison on Memory Usage. The highest score is denoted with an asterisk; the APP score is denoted in bold.

Dataset	Method	1th	2th	3th	4th	5th
Iris	AP	59.0	49.0	164.0	156.0	57.0
	IAPNA	62.0	51.0	15.0*	43.0*	37.0*
	APP	**43.0***	**40.0***	**50.0**	**43.0***	**39.0**
Wine	AP	60.0	55.0	63.0	61.0	65.0
	IAPNA	53.0	24.0*	15.0*	15.0*	70.0
	APP	**39.0***	**40.0**	**41.0**	**39.0**	**41.0**
Car	AP	83.0	88.0	119.0	161.0	154.0
	IAPNA	15.0*	127.0	34.0	166.0	15.0*
	APP	**58.0**	**43.0***	**15.0***	**41.0***	**33.0**
KDD-CUP	AP	103.0	115.0	133.0	142.0	139.0
	IAPNA	167.0	81.0	15.0*	172.0	79.0
	APP	**73.0***	**77.0***	**70.0**	**74.0***	**68.0***

Table 5.9: Uniform-incremental experiment: comparison on the Number of Iterations. The highest score is denoted with an asterisk; the APP score is denoted in bold.

Dataset	Method	1th	2th	3th	4th	5th
Iris$_3$	AP	10.0	8.0	10.0	11.0	12.0
	IAPNA	5.0	6.0	5.0	7.0	9.0
	APP	**4.0***	**3.0***	**3.0***	**3.0***	**2.0***
Wine$_3$	AP	11.0	12.0	12.0	12.0	12.0
	IAPNA	9.0	1.0	2.0	2.0*	2.0
	APP	**4.0***	**2.0***	**3.0***	**2.0***	**3.0***
Car$_4$	AP	27.0	28.0	26.0	31.0	31.0
	IAPNA	25.0	26.0	25.0	29.0	28.0
	APP	**8.0***	**4.0***	**2.0***	**50.0***	**3.0***
KDD-CUP$_{11}$	AP	74.0	82.0	72.0	78.0	84.0
	IAPNA	4.0*	6.0*	6.0*	63.0	72.0
	APP	**26.0**	**21.0**	**18.0**	**16.0***	**20.0***

Table 5.10: Uniform-incremental experiment: comparison on the Number of Clusters. The highest score is denoted with an asterisk; the APP score is denoted in bold. The subscript denotes the number of categories in each dataset.

to the categorical features in such datasets that have been converted to numeric values by using one-hot encoding for vector representation. If we exclude the Wine and the Car dataset, the NMI average score of APP achieves the value of 0.726, while the AP and IAPNA scores are 0.647 and 0.657, respectively. As a further consideration, we note that the best results of APP in terms of NMI are reached on the KDD-CUP dataset where the average score is 0.731, while those of AP and IAPNA are 0.699 and 0.679, respectively. This is a particularly interesting result since KDD-CUP is the dataset with the highest number of objects and categories among those considered.

As a main result, due to the faithfulness property of APP that reduces the number of objects considered for clustering in each time step, we observe that APP is far more scalable than AP and IAPNA in terms of CT, MU, and NI. On average by considering all the time steps and datasets, APP achieves a CT score of 0.083, while AP and IAPNA achieve 8.633 and 14.017, respectively. Also about MU, we note that AP consumes 0.768 MB, while AP and IAPNA consume 39.359 MB and 39.573 MB, respectively. Furthermore, the average NI score of APP is 48.350, while AP and IAPNA obtain the score 101.300 and 62.800, respectively.

According to the above results on the uniform-incremental experiment, we observe that APP is much faster than AP and IAPNA, while consuming much less memory than the two considered baselines. Furthermore, we note that the NC values of APP represent the best approximation among the considered clustering algorithms with respect to the number of categories contained in the datasets. Usually, the NC value of APP is slightly higher and sometimes equal to the number of dataset categories.

Results on the variable-incremental experiment

In the variable-incremental experiment, we performed the same tests of the uniform-incremental experiment on PUR, NMI, CT, MU, NI, and NC. For the sake of simplicity, we report in Table 5.11 only the scores of APP on all the tests and datasets of the variable-incremental experiment. The whole set of results for AP and IAPNA on the variable-incremental experiment is available online. As a general remark, we observe that the

Dataset	**Metric**	**1th**	**2th**	**3th**	**4th**	**5th**
Iris$_3$	PUR	1.000*	0.988*	0.938*	0.897*	0.887*
	NMI	0.616	0.696	0.751*	0.754*	0.718
	CT	0.051	0.048	0.051	0.048	0.058
	MU	0.016*	0.020*	0.025	0.027	0.038
	NI	59.0	45.0	51.0	46.0	50.0
	NC	4.0	4.0	4.0	4.0	5.0
Wine$_3$	PUR	0.816*	0.823*	0.842*	0.834*	0.742*
	NMI	0.412*	0.518*	0.581*	0.604*	0.572*
	CT	0.058	0.044*	0.054	0.047*	0.047*
	MU	0.036*	0.048*	0.057*	0.067	0.079
	NI	44.0*	39.5*	39.5*	37.0*	43.0
	NC	5.0	4.0	4.0	3.0*	5.0
Car$_4$	PUR	0.770*	0.677*	0.578	0.604*	0.535
	NMI	0.364	0.323	0.278*	0.315*	0.213
	CT	0.055*	0.048*	0.037	0.034*	0.032*
	MU	0.046*	0.072	0.088	0.084	0.100
	NI	51.0*	43.0*	46.0	45.0	15.0*
	NC	10.0	11.0	9.0	10.0*	4.0
KDD-CUP$_{11}$	PUR	0.849*	0.838*	0.831*	0.806*	0.744
	NMI	0.719	0.732	0.737	0.732*	0.691
	CT	1.804	1.352	1.500	1.451	1.479
	MU	3.006	3.629	4.054	4.584	5.405
	NI	87.5	67.0*	71.0	72.0*	64.0*
	NC	30.0	28.0	28.0	27.0	25.0

Table 5.11: Variable-incremental experiment: results of APP on all the considered datasets. The asterisks denote the APP scores higher than the corresponding ones in the uniform-incremental experiment.

APP results on the variable-incremental experiment confirm the observations on the uniform-incremental experiment. APP achieves comparable/higher clustering performances than AP and IAPNA algorithms. As a difference with the uniform-incremental experiment, in Table 5.11 we note that the PUR scores for APP are improved. This is in relation to the fact that also a slightly higher number of clusters NC are generated by APP in the variable-incremental experiment.

Ablation study APP is designed to work under the "group evolution" assumption, namely the idea that a new incoming object that differs from past observations is more likely to be considered as an outlier of a previously created cluster rather than as a singleton new cluster. To this end, in the variable-incremental experiment, we inserted a *q* parameter to specify the minimum number of incoming objects per category at a time step *t*.

In the following, we present an ablation study, where the "group evolution" assumption is replaced by an "individual evolution" assumption. In particular, the constraint on the *q* parameter is removed and it is possible that just one or a few objects per category are incoming at a certain time step *t*. The goal of this experiment is to analyze whether and how APP is capable of successfully recognizing the category of incoming objects also when a few elements of that category appear at a certain time step.

In Table 5.12, we show the APP results in terms of PUR and NMI when a minimum number of incoming objects per category *q* is not specified/considered. With respect to the scores on PUR and NMI of Table 5.11, we note that the APP scores are slightly lower on Iris and Car datasets and they are slightly higher on the Wine dataset. We also note that the APP scores on the KDD-CUP dataset are dramatically lower than those shown in Table 5.11.

Metric	Dataset	1th	2th	3th	4th	5th
PUR	Iris	0.923	0.900	0.882	0.882	0.880
	Wine	0.835*	0.881*	0.881*	0.889*	0.888*
	Car	0.702	0.624	0.577	0.602	0.596*
	KDD-CUP99'	**0.586**	**0.182**	**0.165**	**0.135**	**0.410**
NMI	Iris	0.647*	0.677	0.659	0.693	0.640
	Wine	0.481*	0.585*	0.629*	0.642*	0.615*
	Car	0.337	0.280	0.240	0.288	0.300*
	KDD-CUP99'	**0.529**	**0.000**	**0.000**	**0.414**	**0.000**

Table 5.12: Ablation study: PUR and NMI scores of APP when the *q* parameter is not considered and a minimum number of incoming objects per category is not employed. The APP scores that are higher with respect to Table 5.11 are denoted with an asterisk; the scores on the KDD-CUP dataset are denoted in bold.

As a result, we argue that the "group evolution" assumption implemented through the *q* parameter does not significantly affect the APP scores on small datasets like Iris, Car, and Wine where few categories are defined. On the opposite, on large datasets like KDD-CUP where a number of categories are defined, not using the *q* parameter has a strong negative impact on PUR and NMI scores. This means that the "group evolution" assumption implemented through the *q* parameter positively affects the correct recognition of object categories especially when datasets with several categories are considered, while not negatively affecting the PUR and NMI scores on datasets with few categories.

Analysis of clustering results over time As a further test, we consider a specific execution of APP and the related clustering results over six time steps. The goal is to analyze the capability of APP to correctly cluster objects according to the corresponding categories when different incremental schemas are used (i.e., growing, shrinking, stable). In Figure 5.3, we show the results of an APP execution on the Iris dataset. In the dataset, the objects are distinguished in three different categories each one constituted by 50 elements, namely gold-0, gold-1, and gold-2. In the test, the objects of the three categories follow a different incremental schema of arrival. The objects of the gold-0 category are scheduled for arrival according to the stable schema (i.e., 9 gold-0 objects at 0-th and 1-th time steps; 8 gold-0 objects at subsequent time steps). The objects of the gold-1 category follow a shrinking schema focused on time steps from 0-th to 2-th. In particular, 19, 16, and 15 gold-1 objects are scheduled at 0-th, 1-th, and 2-th time steps, respectively. Finally, the objects of the gold-2 category follow a growing schema focused on time steps from 3-th to 5-th. In particular, 12, 13, and 25 gold-2 objects are incoming at 3-th, 4-th, and 5-th time steps, respectively.

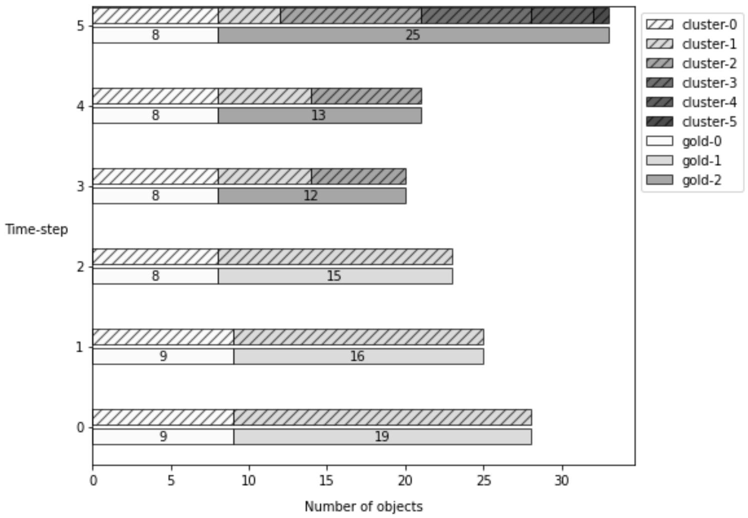


Figure 5.3: Variable-incremental experiment: example of APP results by time step over the Iris dataset.

In Figure 5.3, for each time step, we compare the clusters created by APP against the expected gold clusters based on the category of the incoming objects. We observe that APP works very well in clustering objects of stable and shrinking schema. Indeed, the cluster-0 of APP always succeeds in correctly clustering the gold-0 objects in all the time steps. Similarly, we note that the cluster-1 of APP perfectly reproduces the group of gold-1 objects in all the time steps from 0-th to 2-th where the gold-1 objects are incoming. We also note that some incorrect clustering results are produced by APP on the gold-2 objects that arrive with a growing schema from 3-th to 5-th time steps. In particular, in 3-th and 4-th time steps, the gold-2 objects are distributed in two APP clusters, namely cluster-1 and cluster-2. Cluster-2 represents the APP cluster that better fits to the gold-2 category. A part of the gold-2 objects are wrongly recognized as gold-1 objects and placed in cluster-1. In the 5-th time step, the gold-2 objects are spread over five APP clusters. Again, a (small) part of gold-2 objects are placed in cluster-1 since they are wrongly recognized as gold-1 objects. Coherently with the results of 3-th and 4-th time steps, the cluster-2 of APP seems to be the group that better fits the gold-2 category. The remaining cluster-3, cluster-4, and cluster-5 represent noisy groups with respect to the expected gold categories of Iris. According to the above observations, we argue that clustering errors mostly occur when the incoming objects follow a growing incremental schema. This is due to the fact that the new category appears with a low number of objects in the first time step and this schema challenges the correct recognition of the new cluster to create.

5.6 Discussion and considerations

In this chapter, we propose A-Posteriori affinity Propagation (APP) as an extension of Affinity Propagation (AP). APP is conceived to work in incremental scenarios by enforcing faithfulness and forgetfulness through cluster consolidation/stratification. Evaluation results on popular benchmark datasets are provided to assess the performance of APP in two different incremental settings. The results show that APP obtains comparable results on cluster quality with respect to AP and IAPNA algorithms, while achieving high scalability performances at the same time. Our results show that APP is suitable for application scenarios where the "group evolution" assumption holds.

However, it is important to consider some limitations when interpreting our evaluation. Specifically, while we thoroughly evaluated APP against popular benchmarks, we did not assess its performance in real case-study datasets that might better represent real-world application scenarios. Since this thesis focuses on modeling semantic change, we limited our first evaluation of APP to these benchmarks. *We will further expand and illustrate the applicability of APP for LSC in the next chapter.*

Moreover, a more comprehensive evaluation for general real-world scenarios should involve benchmarking APP against other evolutionary clustering algorithms. In our evaluation, we considered only AP extensions, as AP is generally regarded as an established baseline in word meaning modeling. Specifically, we compared APP only with the standard AP and the incremental IAPNA, as other evolutionary AP extensions lack official implementations available for evaluation.

Chapter 6

The *What is Done is Done* approach

> "*How now, my lord! Why do you keep alone,*
> *Of sorriest fancies your companions making,*
> *Using those thoughts which should indeed have died*
> *With them they think on? Things without all remedy*
> *Should be without regard. What's done is done.*"
>
> William Shakespeare, *Macbeth*

6.1 Introduction

In the previous chapter, we proposed a novel clustering algorithm called *A-Posteriori affinity Propagation* (APP) and evaluated its effectiveness against standard clustering benchmarks. We now turn our attention to its potential application in LSC. In this chapter, we employ APP to incrementally cluster word embeddings, aiming to capture semantic change and the evolution of word meanings across a diachronic corpus.

Initially, we presented this *sense*-based approach to LSC at the 3rd Workshop on Computational Approaches to Historical Language Change (Tahmasebi et al., 2022c). We originally referred to this approach as *What is Done is Done* (WiDiD, Periti et al., 2022). The idea underlying WiDiD is that the word contexts observed in the past are consolidated as a set of clusters that constitute the "memory" of the word meanings observed so far. Such a memory is exploited as a basis for subsequent word observations, so that the meanings observed in the present are stratified over the past ones. In particular, the idea of WiDiD is that the clusters of word meanings previously created cannot be changed (*what is done is done*), and the word meanings that are observed in the present must be stratified/integrated over the past ones. In each consecutive time period, the word embeddings of that time period are compared to the already existing clusters. They either get assigned to an existing cluster or are allowed to form a new cluster, and thus the memory gets updated at each time period. As a result, the stratified layers of clusters over time allow assessment of the quantity of semantic change as well as reconstruction of the evolution of a word's meanings.

Chapter outline.

This chapter includes materials originally published in the following publications:

> Francesco Periti, Alfio Ferrara, Stefano Montanelli, and Martin Ruskov. 2022. What is Done is Done: an Incremental Approach to Semantic Shift Detection. In Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change, pages 33–43, Dublin, Ireland. Association for Computational Linguistics.

> Francesco Periti, Sergio Picascia, Stefano Montanelli, Alfio Ferrara, and Nina Tahmasebi. 2024**e**. Studying Word Meaning Evolution through Incremental Semantic Shift Detection. Language Resources and Evaluation.

> Silvana Castano, Alfio Ferrara, Stefano Montanelli, and Francesco Periti. 2024. Incremental Affinity Propagation based on Cluster Consolidation and Stratification. eprint 2401.14439, arXiv. Under review.

This chapter is organized as follows. In Section 6.2, we present the WiDiD approach for LSC. In Section 6.3, we expand the discussion on the set of techniques employed by WiDiD for analyzing and detecting semantic change. In Section 6.4, we illustrate two exemplary applications of WiDiD in real-world scenarios. In Section 6.5, we evaluate WiDiD over seven LSC benchmarks across multiple languages. Empirical results show that WiDiD is at least comparable to state-of-the-art approaches, while outperforming the state-of-the-art for certain languages. Finally, in Section 6.6, we discuss the use of APP for LSC by examining both its benefits and drawbacks.

6.2 WiDiD: What is Done is Done

Consider a dynamic, diachronic document corpus $C = \bigcup_{t=0} C^t$ where C^t denotes a set of documents added at time t^i. Given a target word w, our goal is to analyze how the meaning(s) of w changed along C. Documents in C are considered as a data stream segmented into a sequence of time periods. As shown in Figure 6.1, WiDiD consists of a four-step pipeline that is repeatedly applied to the progressively added documents in C: **1)** *Document Selection*, **2)** *Embedding Extraction*, **3)** *Incremental Clustering*, **4)** *Clustering Analysis*.

At the first time step (i.e., $t = 0$), only the documents in C^0 are considered. As a result, only a *synchronic* analysis of clustering is possible, as there is no knowledge available about the meaning of w in the past. Then, for each subsequent step $t = 1...n$, the knowledge of the w meaning(s) detected in the past time periods (i.e., time periods $0...t-1$) is exploited by the step **3)** to cluster the documents in C^t. This *diachronic* analysis of clustering can provide insights into the semantic change that has occurred.

Notation	Definition
w	Target word
C^t	Set of documents at time t
C^t_w	Subset of documents of C^t containing the word w
$e^t_{w,i}$	Embedding of the word w in the i-th document of C^t_w
Φ^t_w	Set of the embeddings of w in the corpus C^t_w
K^t_w	Set of clusters obtained at the t-th iteration for w
$\phi_{w,k}$	k-th cluster containing the embeddings of the word w
$\phi^t_{w,k}$	Subset of embeddings from time t in the cluster $\phi_{w,k}$
$\mu^t_{w,k}$	Prototypical representation of w for $\phi^t_{w,k}$
M^t_w	Set of prototypes $\mu^t_{w,k}$ available at time t
π^t_w	Polysemy of the word w at time t
S^t_w	Semantic shift of the word w at time t
$\rho^t_{w,k}$	Prominence of the cluster $\phi^t_{w,k}$ at time t
$\mathcal{T}^t_{w,k}$	Sense shift of the cluster $\psi_{w,k}$ at time t

Table 6.1: A reference table of notation used in the chapter.

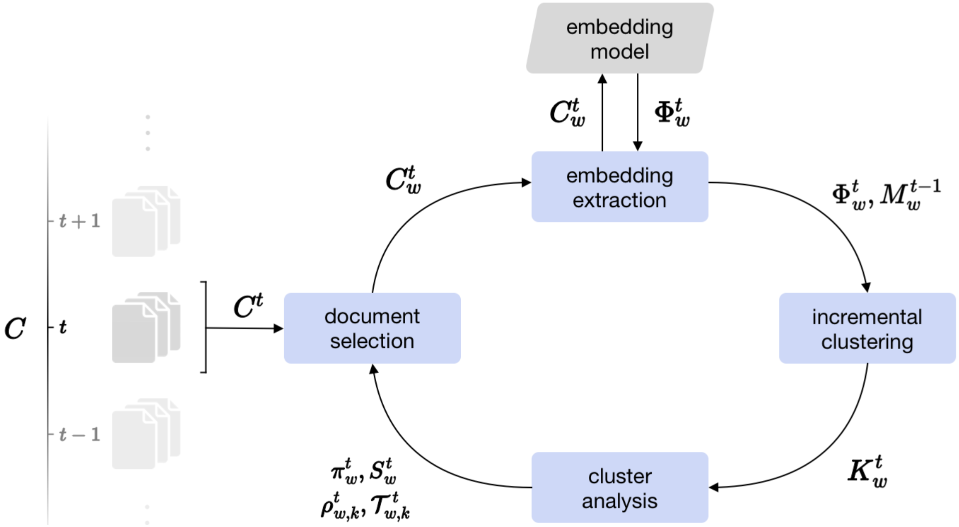


Figure 6.1: WiDiD: an incremental approach to LSC.

The documents in C^t are processed via WiDiD as follows. For the sake of clarity, the notation used throughout this chapter is summarized in Table 6.1.

Document Selection (DS). In this step, WiDiD selects the subset of documents $C^t_w \subseteq C_t$ that contains an occurrence of the word w. Since semantic change is often accompanied by morphosyntactic drift (Kutuzov et al., 2021a), we consider any derived form of the lemma of w (e.g., plural) as an occurrence of w.

Embedding Extraction (EE). In this step, WiDiD encodes each occurrence of the target word w in C^t_w with a different representation. Because currently, contextualized embeddings represent the preferred tool for addressing SSD (Periti and Montanelli, 2024), we will use embeddings generated by standard BERT-like models (i.e., BERT, mBERT, XLM-R). The WiDiD approach is however more general and can be applied regardless of the specific model used to represent individual word occurrences.

In particular, to extract contextualized embeddings for a specific target word w, we fed the considered model with individual text sequences containing an occurrence of w. For each occurrence of w, we extracted a contextualized embedding from the last hidden layer of the model. Due to the byte-pair input encoding scheme employed by BERT models, some word occurrences may not correspond to words but rather to word pieces (Sennrich et al., 2016). Therefore, if a word was split into more than one sub-word, we built a single word embedding by averaging the corresponding sub-word embeddings. The final output of this step is the set Φ^t_w containing all the embeddings of the word w generated for the corpus C^t. Formally,

$$\Phi^t_w = \{e^t_{w,1}, \dots, e^t_{w,m}\} ,$$

where $e^t_{w,j}$ is the embedding of w in the j-th document and m is the number of documents in C^t_w.

Incremental Clustering (IC). WiDiD first ($t = 0$) uses the standard AP algorithm over Φ^0_w. This results in a set of clusters denoted as K^0_w. For $t > 0$, clustering is performed using the APP algorithm to cluster the embeddings Φ^t_w in groups representing *sense nodules*, “lumps of (word) **meaning** with greater stability under contextual changes” (Kutuzov et al., 2022b). We denote the set of resulting clusters as K^t_w. At each time step, APP creates an additional *sense prototype* embedding $\mu^{t-1}_{w,k}$ for each cluster $k \in K^{t-1}_w$ by averaging all its enclosed embeddings, meaning that $\mu^{t-1}_{w,k}$ is the centroid of the k-th cluster. The resulting sense prototypes constitute the “memory” of the word meanings observed so far. This memory is then exploited as the basis for subsequent word observations in the current time period. In particular, we denote as M^{t-1}_w the set of sense prototypes $\mu^{t-1}_{w,k}$ available at time $t-1$. Hence, APP consists of performing the standard AP over the set of embeddings $\Phi^t_w \cup M^{t-1}_w$. As a final step of APP, each sense prototype $\mu^{t-1}_{w,k}$ is removed, and the original embeddings compressed into $\mu^{t-1}_{w,k}$ are assigned to its corresponding cluster. This ensures that all the embeddings associated with a sense prototype at time $t-1$ are grouped together within the same cluster at the time t. This way, clusters of word meanings previously created cannot be changed, and the word meanings that are observed in the present must be stratified/integrated over the past ones.

Incremental clustering represents a significantly more scalable solution than existing approaches (Montariol et al., 2021; Kanjirangat et al., 2020). Since clusters formed in previous steps are considered as unique prototypes, in each clustering step we work with a significantly smaller set of embeddings, while at the same time eliminating the need for cluster alignment techniques.

Clustering analysis (CA) In this step of WiDiD, each clustering result obtained as an IC output is analyzed to interpret the meaning of words from both a synchronic and diachronic perspective. This step of WiDiD is presented in further detail in Section 6.3, where we introduce a comprehensive set of metrics specifically designed to describe both a target word and its sense nodules over time.

6.3 Cluster analysis

For each time period t, the incremental clustering (IC) results in a set of k clusters $K^t_w = \phi_{w,1}, ..., \phi_{w,k}$. In particular, we denote the set of embeddings from Φ^t_w enclosed in the k-th cluster as $\phi^t_{w,k}$. Formally, we define $\phi^t_{w,k} = \phi_{w,k} \cap \Phi^t_w$. This implies that $\phi^t_{w,k} \subset \Phi^t_w$ is the subset of embeddings extracted at time t that are members of the cluster $\phi_{w,k}$ during that specific time step.

To be able to analyze the sequence of clustering results for a word w, we propose a set of metrics that characterize w both from a synchronic and diachronic perspective. Regardless of the perspective, these metrics are also conceived to inspect a particular clustering result by considering two linguistic targets:

1. *word*: when all clusters are considered overall, we analyze the target word w;
2. *sense nodules*: each cluster is considered individually. Ideally, when focusing on a target cluster, our aim is to analyze the particular word meaning associated with that cluster. However, since clusters are derived from vector representations generated by distributional models, each cluster loosely represents a sense of the word w. As a result, when considering a cluster individually, our analysis centers on a specific *sense nodules* or *cluster of corpus usage.* (Kutuzov et al., 2022b).

6.3.1 Synchronic perspective

From a synchronic perspective, words and sense nodules are considered within a specific time period, without taking into account their evolution in meaning. We define two metrics to describe the status of words and sense nodules, respectively.

Polysemy, denoted as π^t_w, describes the status of a word at a particular time period t. Polysemy is defined as the number of "active" sense nodules present at time t, i.e., sense nodules from earlier periods integrated with new elements as well as newly identified sense nodules. Intuitively, the more clusters there are, the more polysemous the word is.

$$\pi^t_w = |K^t_w| \tag{6.1}$$

Prominence, denoted as $\rho^t_{w,k}$, describes the status of a sense nodule at a particular time period t. Prominence is defined as the prevalence of an active sense $\phi^t_{w,k}$ at time t relative to the other active sense nodules. Intuitively, the more members in a cluster, the more prominent the sense nodule is.

$$\rho^t_{w,k} = \frac{|\phi^t_{w,k}|}{|\Phi^t_w|} \tag{6.2}$$

6.3.2 Diachronic perspective

From a diachronic perspective, words and sense nodules are considered across time periods, taking into account their evolution in meaning. The clusters at the last iteration are used in the analysis and are traced over time, thus avoiding a complex analysis of potential mergers across all time periods. We define two metrics to describe the evolution of words and sense nodules, respectively.

Semantic shift, denoted as $\mathcal{S}_w$, describes the degree of lexical semantic change of a word over two consecutive time periods. Semantic shift is defined as the degree of dissimilarity in the prominence of active sense nodules between these time periods. Intuitively, the greater the dissimilarity between time periods t and $t-1$, the higher the degree of semantic shift a word has undergone. Following Giulianelli et al. (2020), we formally define semantic shift as the Jensen-Shannon divergence (JSD) over the prominence distributions P_w^{t-1} and P_w^t, where the $k-$th value of a distribution P_w^i is the prominence $\rho_{w,k}^i$ associated with the $k-$th sense nodule resulting from the last enforced clustering step.

$$JSD(P_w^{t-1}, P_w^t) = \frac{1}{2}\left(KL(P_w^{t-1}||M) + KL(P_w^t||M)\right) \ ,$$

where $M = (P_w^{t-1} + P_w^t)/2$, and KL represents the Kullback-Leibler divergence, as JSD is a symmetrization of KL.

Sense shift, denoted as $\mathcal{T}_{w,k}$, describes the degree of lexical semantic change of a specific word's sense nodule over two consecutive time periods. Sense shift is defined as the degree of distance in the sense prototypes $\mu_{w,k}^t$ and $\mu_{w,k}^{t-1}$ for these time periods. Intuitively, the greater the difference between time periods t and $t-1$, the greater the degree of sense shift a sense nodule undergoes. Unlike $\mathcal{S}_w$, $\mathcal{T}_{w,k}$ aims to capture lexical semantic change specific to sense nodules. This score quantifies how a cluster changes over time, aiding in the identification of semantic changes other than sense loss and acquisition (e.g., amelioration, pejoration, broadening, or narrowing).

We formally define the sense shift of the $k-$th sense nodule as the cosine distance between the sense prototypes $\mu_{w,k}^t$ and $\mu_{w,k}^{t-1}$.

$$\mathcal{T}_{w,k}(\mu_{w,k}^t, \mu_{w,k}^{t-1}) = \frac{\mu_{w,k}^t \cdot \mu_{w,k}^{t-1}}{||\mu_{w,k}^t||\ ||\mu_{w,k}^{t-1}||}$$

6.3.3 Clustering visualization

To facilitate the analysis and interpretation of the evolution of a word's meaning, we propose a new visualization that supports the synchronic and diachronic metrics enforced in cluster analysis. Unlike the visualization methods for diachronic semantic change presented in Kazi et al. (2022), this visualization is particularly suited to a posteriori analysis (see Section 6.3.1) of the last clustering result of WiDiD. Our visualization provides valuable insights into the different sets of sense nodules held by a word over time, as well as clearly

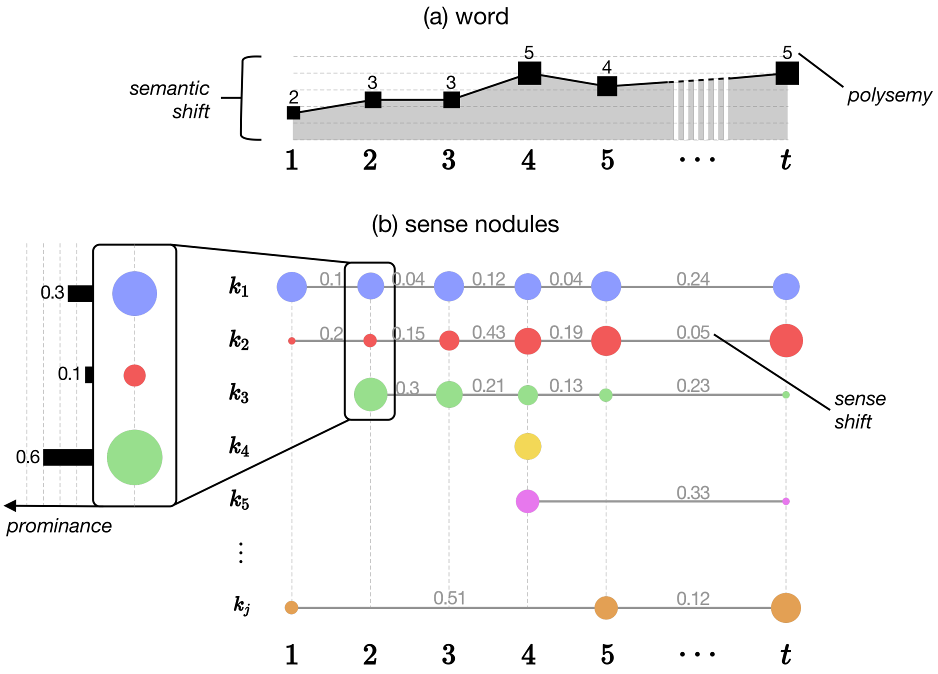


Figure 6.2: Clustering visualization: prototype visualization of word meaning evolution. Subfigure (a) represents the polysemy and semantic shift of a word over time. Subfigure (b) represents the prominence and sense shift of the sense nodules of that word over time.

representing the evolution of those sense nodules.

For the sake of clarity, we describe the rationale of the visualization by considering the prototype of an arbitrary word w illustrated in Figure 6.2. The figure consists of two subfigures (a) and (b), representing the synchronic and diachronic metrics for (a) a target word and (b) its sense nodule, respectively. In both subfigures, the x -axis represents time.

In subfigure (a), each square represents a snapshot of a specific word at a particular time period t. The size of each square reflects the polysemy π^t_w of the word at time t. Semantic shift values over time are reported on the y-axis.

In subfigure (b), each circle in the figure represents a snapshot of a specific sense nodule at a particular time period t. The evolution of different sense nodules (i.e., k_1, ..., k_j) is illustrated on the y-axis using different colors. Intuitively, the presence/absence of a circle at time t indicates the active/inactive state of the related sense nodule. The size of each circle reflects the prominence ρ^t_w of the corresponding sense nodule at time t. Sense shift values over time are reported on the links connecting the snapshots of sense nodules with their respective immediately subsequent snapshots.

6.4 Real applications of WiDiD

We now report on two practical applications of WiDiD.

1. The first application is presented in Section 6.4.1 and involves a large corpus of Vatican publications from 1431 to 2020. This application was originally presented in Castano et al. (2024), when the cluster visualization techniques were still in development. It serves as an illustrative example of potential mergers due to cluster *consolidation* and *stratification* across consecutive time periods (see Chapter 5).

2. The second application is presented in Section 6.4.2 and involves a large corpus of Italian parliamentary speeches from 1948 to 2020. This application was originally presented in Periti et al. (2024e). It complements the preceding one and investigates, through the cluster *visualization* techniques introduced earlier in this chapter, the clusters obtained in the final APP iteration to trace sense evolution over time.

The domains of these applications represent relevant cases for detecting semantic change, as they concern prominent issues in public and social contexts. Our main goal is to demonstrate a practical LSC application of WiDiD to trace the evolution of clusters over time. Hence, the APP pruning threshold th_γ is set to ∞, as our experiment aims to focus on cluster evolution over time rather than analyze the effects of the forgetfulness property on irrelevant clusters. Although a quantitative evaluation is not possible due to the lack of an annotated benchmark (i.e., gold scores for a set of target words), we provide a qualitative analysis of the results to assess the effectiveness of WiDiD in LSC.

In both applications, the first sub-corpus is used in the initial run of AP, followed by the incremental addition of the remaining sub-corpora in subsequent APP iterations.

6.4.1 Vatican publications

Setup. In this application, we consider a corpus of Vatican publications. Our corpus contains 29k documents extracted from the digital archive of the Vatican website and it consists of all the web-available documents, spanning from the papacy of Eugene IV to Francis (1431-2023). Although the documents are available in various languages, including Italian, Latin, English, Spanish, and German, we downloaded the Italian corpus since the largest number of documents are available in this language.

To set-up this illustrative application, we first define a target word w we aim to detect its semantic change within the Vatican corpus. Then, we split the corpus into six sub-corpora, each one denoting a specific time period. It is worth noting that for most of the earlier pontificates, a few documents are available (e.g., Eugene IV) or none at all (e.g., Nicholas V). To address the skewed distribution of documents over time, we aggregated popes and related documents to ensure that each sub-corpus contains at least 50 occurrences of the target word w. Furthermore, we performed a random sampling of 100 occurrences of w from each sub-corpus when more occurrences are available to ensure that the number of occurrences are comparable across the sub-corpora.

We exploit the Italian pre-trained BERT model (i.e., *bert-base-italian-cased*) to represent each occurrence of the target word w as a word embedding vector.

As a target word, we consider w = novità (**novelty**). The Vatican corpus is split into the following sub-corpora: *before Leo XIII*, with documents prior to 1878; *from Leo XIII to Pius XI*, with documents in the range 1878–1939; *from Pius XII to John XXIII*, with documents in the range 1939–1963; *Paul VI*, with documents in the range 1963–1978; *Benedict XVI*, with documents in the range 2005–2013; *Francis I*, with documents up to 2023. It is worth noting that we do not include the pontificate of John Paul II in this analysis. The richness and the variety of documents of John Paul II is significantly higher than the other pontificates and we note that it has been used in several different contexts and meanings, thus introducing a really challenging LSC task. So, we decided to exclude the documents of John Paul II since the goal of our application is to show the behavior of WiDiD on cluster evolution and not to discuss the WiDiD effectiveness on a custom LSC task.

Results. In Figure 6.3, we provide an example of cluster evolution according to the stratification criteria presented in Chapter 5. Each cluster contains a set of contextual embeddings of the target word novelty and it denotes a corresponding meaning of novelty at a certain time by considering the documents of the Vatican corpus until that moment.

A cluster k is represented as a box with an associated identifier. The cluster size denotes the cumulative number of elements in the cluster at each iteration: the larger the cluster box, the greater the number of cluster elements. In the example, we use the same cluster identifier across different iterations when the cluster is the result of a *stratification-by-enrichment*, while we assign new identifiers to clusters resulting from *stratification-by-creation* and *stratification-by-merge*.

The example of Figure 6.3 shows that just one meaning of the word novelty could be recognized in the $1st$ WiDiD iteration; and further meanings appeared in subsequent executions, especially in the iterations from $4th$ to $6th$, where the use of the word novelty becomes strongly polysemous.

The cluster k0 in the $1st$ WiDiD iteration is an example of *stratification-by-creation* and it describes the use of the word novelty as a negative, dangerous concept, since new ideas and novel practices were considered as a threat to the traditional teachings of the Church by the earlier pontificates. The cluster k0 is populated with new elements in the $2nd$ iteration (*stratification-by-enrichment*), when a new cluster k1 is also introduced with embeddings of the novelty occurrences from the documents of the $2nd$ sub-corpus (*stratification-by-creation*). The clusters k0 and k1 are joined in the $3rd$ iteration to generate the cluster k2 (*stratification-by-merge*). The cluster k2 remains unchanged in subsequent iterations from $4th$ to $6th$ (no more documents are found similar to k2), confirming that such a conservative, right-wing position of the Church has been abandoned after the Second Vatican Council (1962–1965).

In this example, the clusters k0–k2 are equipped with a textual description that has the goal to summarize the cluster contents and the related meaning of the word novelty in the cluster. Since cluster labeling is not the focus of this study, we leverage ChatGPT[1] to generate the cluster summaries of our examples. To label a

[1] https://openai.com/blog/chatgpt/

cluster, we collect the text sources in the Vatican corpus that are associated with the occurrences of the word novelty in the cluster and we ask ChatGPT to summarize the common topic.

As a further example, in Figure 6.4, we show the evolution/stratification over time of those clusters that are finally merged into the cluster k26 at the $6th$ iteration of WiDiD in Figure 6.3. The example of Figure 6.4 is about the usage of the word novelty in relation to societal, cultural, and religious change. In particular, we focus on the period from 1939 to 2023 (iterations from $3rd$ to $6th$), although this meaning of novelty appeared in the $2nd$ iteration with the clusters k3 and k4 as examples of *stratification-by-creation*. According to Figure 6.3, the $3rd$ iteration is characterized by the emergence of new relevant clusters such as k5 and k6 through *stratification-by-creation*, while the cluster k3 increases its importance with new elements through *stratification-by-enrichment*. The cluster k4 remains unchanged, and a new marginal cluster called k7 is created. In the $4th$ iteration, the number of clusters about this meaning of novelty is strongly increased (*stratification-by-creation*), probably due to the dynamism of ideas introduced by the Second Vatican Council and reflected in the Vatican documents. Such a variety of positions at the $4th$ iteration is represented in Figure 6.4 by the clusters k6, k8, and k17. The $5th$ iteration is mostly characterized by *stratification-by-merge* operations and the clusters k20, k21, and k22 represent the main result of WiDiD on this meaning of novelty. About the cluster k21, we note that it is the result of a merge operation that involves a number of clusters of the previous iteration (i.e., the $4th$ one), and it is also strongly increased in importance due to the insertion of several elements (i.e., novelty occurrences) of the current $5th$ iteration.

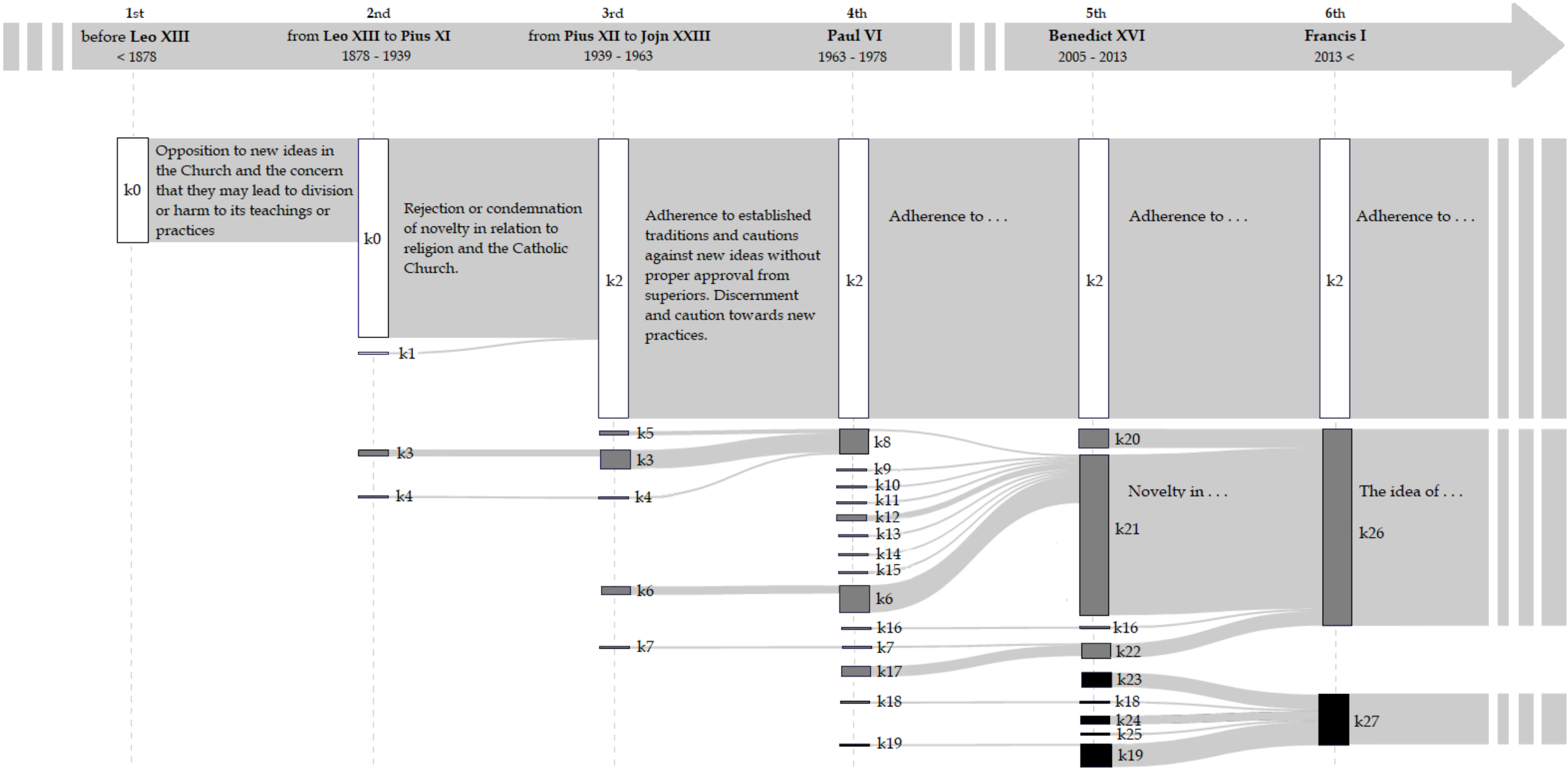
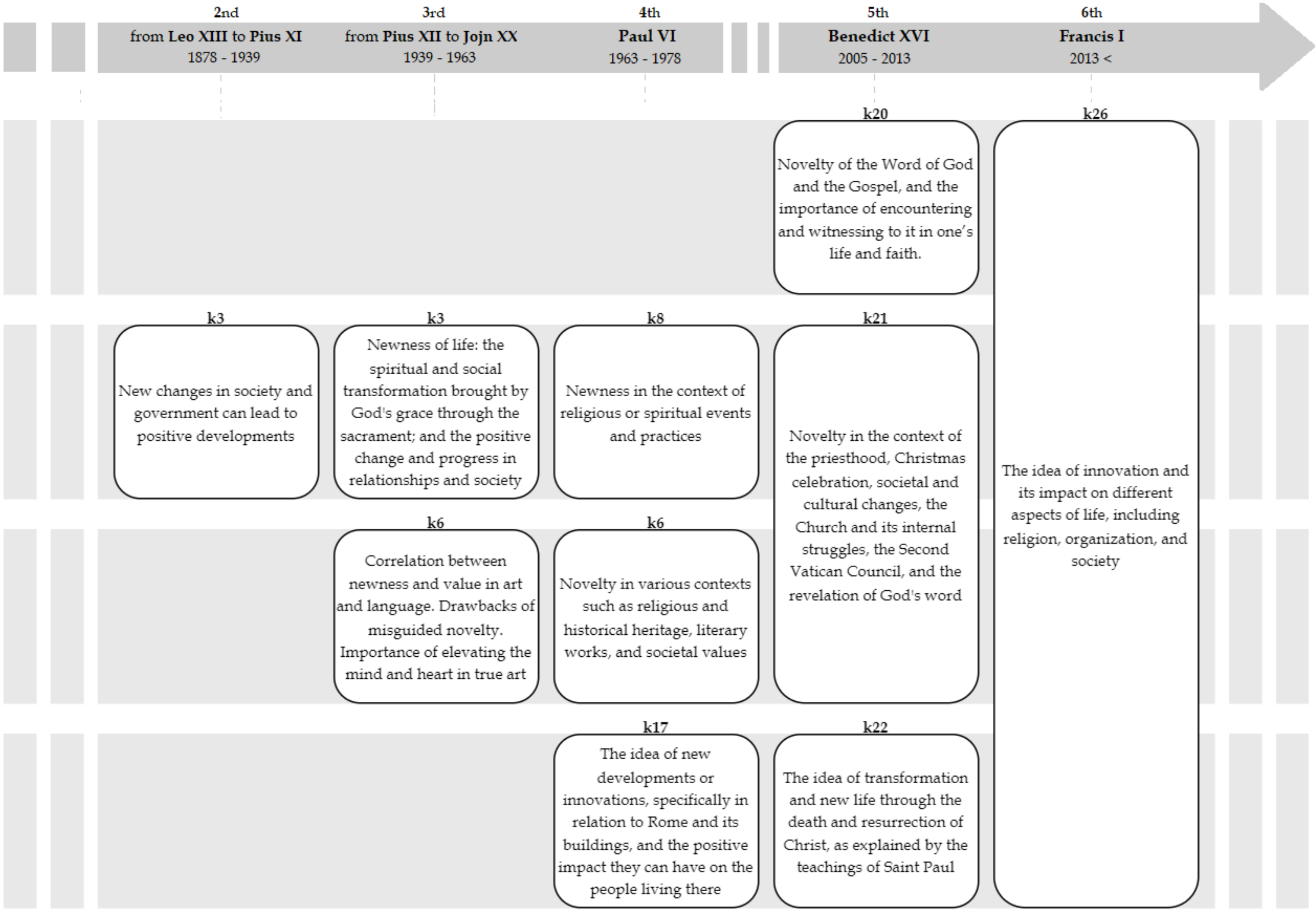

1st
before Leo XIII
< 1878
2nd
from Leo XIII to Pius XI
1878 - 1939
3rd
from Pius XII to Jojn XXIII
1939 - 1963
4th
Paul VI
1963 - 1978
5th
Benedict XVI
2005 - 2013
6th
Francis I
2013 <
k0
Opposition to new ideas in the Church and the concern that they may lead to division or harm to its teachings or practices
k0
Rejection or condemnation of novelty in relation to religion and the Catholic Church.
k1
k3
k4
k2
Adherence to established traditions and cautions against new ideas without proper approval from superiors. Discernment and caution towards new practices.
k5
k3
k4
k6
k7
k2
Adherence to . . .
k8
k9
k10
k11
k12
k13
k14
k15
k6
k16
k7
k17
k18
k19
k2
Adherence to . . .
k20
Novelty in . . .
k21
k16
k22
k23
k18
k24
k25
k19
k2
Adherence to . . .
The idea of . . .
k26
k27

Figure 6.3: The WiDiD application on the Vatican corpus for the word novelty.



k20: Novelty of the Word of God and the Gospel, and the importance of encountering and witnessing to it in one's life and faith.

k26: The idea of innovation and its impact on different aspects of life, including religion, organization, and society

k3 (2nd): New changes in society and government can lead to positive developments

k3 (3rd): Newness of life: the spiritual and social transformation brought by God's grace through the sacrament; and the positive change and progress in relationships and society

k8: Newness in the context of religious or spiritual events and practices

k21: Novelty in the context of the priesthood, Christmas celebration, societal and cultural changes, the Church and its internal struggles, the Second Vatican Council, and the revelation of God's word

k6 (3rd): Correlation between newness and value in art and language. Drawbacks of misguided novelty. Importance of elevating the mind and heart in true art

k6 (4th): Novelty in various contexts such as religious and historical heritage, literary works, and societal values

k17: The idea of new developments or innovations, specifically in relation to Rome and its buildings, and the positive impact they can have on the people living there

k22: The idea of transformation and new life through the death and resurrection of Christ, as explained by the teachings of Saint Paul

Figure 6.4: The evolution/stratification of clusters that are finally merged into the cluster k26 of Figure 6.3. For the sake of readability, the cluster description is provided only for k3, k6, k8, k17, k20, k21, k22, k26.

The result at the $5th$ iteration also includes the (minor) cluster k16 that remains unchanged with respect to the previous iteration (no elements of the $5th$ iteration are inserted in this cluster). The summary descriptions of clusters k20, k21, and k22 are provided in Figure 6.4. This meaning of novelty is finally reconciled in a unique cluster k26 at the $6th$ iteration through a final *stratification-by-merge* operation.

A final example of evolution/stratification is provided in Figure 6.5 about the clusters k19, k23, and k27 of Figure 6.3. This example is about the usage of novelty in relation to the innovation of Christianity, a new understanding of the Church's teaching, and effects on the followers. In this example, we focus on the $5th$ and $6th$ iterations where most of the clusters about this meaning of novelty appear, thus highlighting the very recent emergence of such a discussion in the Church debate. In Figure 6.5, we show the descriptions of clusters k19 and k23 that are the most representative at the $5th$ iteration and that are finally merged into cluster k27 at the $6th$ iteration.

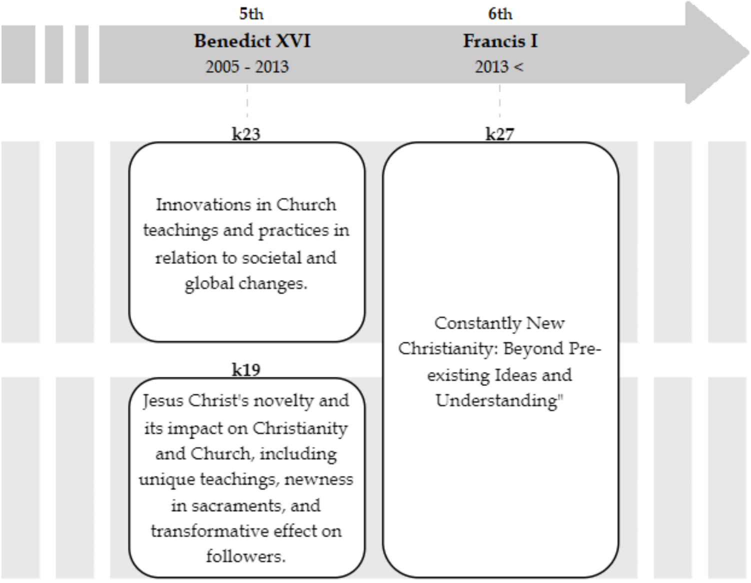

5th
Benedict XVI
2005 - 2013
6th
Francis I
2013 <
k23
Innovations in Church teachings and practices in relation to societal and global changes.
k27
Constantly New Christianity: Beyond Pre-existing Ideas and Understanding"
k19
Jesus Christ's novelty and its impact on Christianity and Church, including unique teachings, newness in sacraments, and transformative effect on followers.

Figure 6.5: The evolution/stratification of clusters k19, k23, and k26 of Figure 6.3.

It is worth stressing that WiDiD allows to represent all the various meaning/interpretations associated with the word novelty at each iteration. Furthermore, the stratification criteria are able to track the transformations of clusters over time, as well as to reconcile all the branches of a certain meaning into a summary cluster at the last iteration, thus providing a convenient picture to the scholar/analyst that aims to explore the evolution of novelty in the whole Vatican corpus.

6.4.2 Parliamentary speeches from the Italian Chamber of Deputies

Setup. In this application, we consider a corpus of parliamentary speeches from the Italian Chamber of Deputies. Our corpus spans a period of 72 years, from the 1st legislature of the Italian Republic after the Constituent Assembly (1948) to February of the 18th Republican Legislature (2020). This corpus was created by collecting all the available plenary session transcripts at the time of downloading from the Italian Parliament website[2].

	Time periods																	
Legislature	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Start date	1948	1953	1958	1963	1968	1972	1976	1979	1983	1987	1992	1994	1996	2001	2006	2008	2013	2018
End date	1953	1958	1963	1968	1972	1976	1979	1983	1987	1992	1994	1996	2001	2006	2008	2013	2018	2020
# Tokens	13.0 M	13.8 M	18.3 M	18.6 M	10.1 M	8.0 M	6.0 M	11.7 M	9.6 M	11.3 M	5.2 M	4.5 M	12.8 M	12.3 M	4.3 M	12.4 M	14.3 M	5.5 M

Table 6.2: Summary of the case study corpus of Italian Parliamentary speeches.

To set-up this WiDiD application, we first define a set of target words we aim to detect its semantic change within the Italian parliamentary corpus. Since the corpus was produced by OCR scanning, it included numerous spurious characters where words had been incorrectly recognized and introduced into the text, degrading the quality of the data. To address this issue, we performed an additional processing step to exclude speech with purely procedural content (e.g., *The MP* [SURNAME NAME] *asks to speak*) and filtered out speech associated with a high level of noise (e.g., spurious characters and other artifacts introduced during the OCR scanning process). To enhance scalability in this study, we reduced the number of embeddings to store and process by randomly sampling a fixed number of occurrences of each target word (i.e., 100).

We exploit the Italian pre-trained BERT model (i.e., *bert-base-multilingual-cased*) to represent each occurrence of the target word w as a word embedding vector. Although we initially experimented with a monolingual pre-trained BERT model (*bert-base-italian-uncased*), the empirical results revealed poor quality. Empirical results obtained with the multilingual model indicated a higher level of quality. We hypothesize that multilingual models can leverage their larger, cross-lingual contextualization and pre-trained knowledge to better handle the various text quality issues present in our OCR-corrupted data.

For the sake of simplicity, we consider $w = \mathsf{pulityo}$ (**clean**) as the main target word. We thus provide only a few illustrative examples for other words. However, the comprehensive list of words, including their polysemy and semantic shifts as well as their sense nodules with associated prominence and sense shifts, are available online for further reference.

The legislatures provide a natural criterion for splitting the corpus over time, meaning that a separate sub-corpus C_i is defined for each legislature i (see Table 6.2).

Manually examining sentences in a specific cluster to interpret the clusters and the semantic change between two time periods is laborious and time-consuming. It involves a meticulous process of close-reading because multiple sentences are present within each cluster. Thus, like Montariol et al. (2021), we automatically extracted the most discriminating words for each cluster to minimize human effort. In particular, we first lemmatized each sentence within the clusters. Then, we treated each cluster as an individual document

[2] https://dati.camera.it/it/dati/

and considered all the clusters as a corpus. For each cluster, we calculated the Term Frequency-Inverse Document Frequency (TF-IDF) score of every word. To ensure the selection of the most meaningful keywords, we eliminated stopwords and excluded parts of speech other than nouns, verbs, and adjectives. Thus, we obtained a ranked list of keywords for each cluster, and the top-ranked keywords were then used for cluster interpretation. Similar to the previous application, we also leverageg ChatGPT to generate the cluster summaries of our examples.

Note that recent work has demonstrated that the geometry of BERT's embedding space exhibits anisotropy, meaning that the contextualized embeddings occupy a narrow cone within the vector space, leading to very small values of cosine distance (Ethayarajh, 2019). Thus, for the sake of readability, we normalized the shift scores of our experiment by the maximum shift value we obtained.

Results. As an example, Figure 6.6 (a) and 6.6 (b) are a visual representation of the result of the cluster analysis for the Italian word `pulito` (*clean*). This word holds particular significance in the Italian context as it represents an adjective commonly associated with cleanliness. However, it gained a specific historical connotation during the early '90s owing to its association with the fight against corruption.

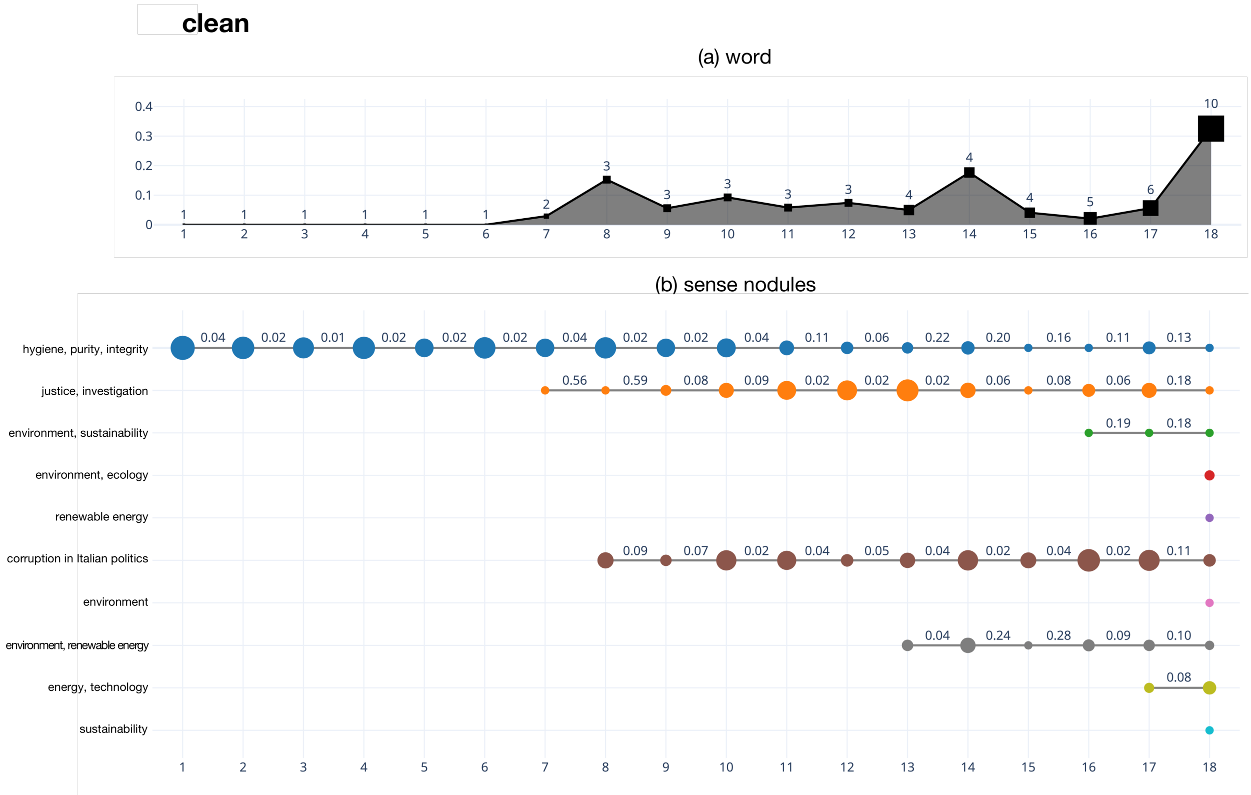


Figure 6.6: Clustering visualization: (a) *semantic shift* and *polysemy* of the Italian word "pulito" (e.g., *clean*); (b) *sense shift* and *prominence* of the sense nodules of the Italian word "pulito" (e.g., *clean*).

Figure 6.6 (a) summarizes Figure 6.6 (b), providing insights into the polysemy of the word and its overall semantic shift across different time periods. The greatest semantic shifts occur in the time intervals 7–8, 13–14, and 17–18. The first time interval is associated with the acquisition of a new sense nodule (i.e., *corruption in Italian politics*). The second time interval is associated with a change in the distribution of sense nodule prominence; for example, in the 14th legislature, the sense nodule *environment, renewable energy* exhibits its maximum prominence. The third time interval is characterized by the emergence of several new sense nodules. Interestingly, the algorithm validates our expectations by capturing the emergence of new sense nodules related to the environment and renewable energy. Indeed, recent years have shown increasing global attention to environmental issues due to factors such as concerns about climate change.

In the discussion of Figure 6.6 (b) we adopt the ecological view of word change proposed by Hu et al. (2019). They suggest that word sense nodules can compete for dominance and cooperate for mutual benefit (i.e., remain active), similar to organisms in an ecosystem. As a complementary view of Figure 6.6, Table 6.3 shows the proportion of documents (i.e., prominence) assigned to each sense nodule.

The cluster analysis in Figure 6.6 (b) captures examples of semantic shifts of the word over time. For instance, we observe an *evergreen* sense nodule (i.e., always present across all considered time periods) associated with the label *hygiene, purity, and integrity*. This sense nodule represents the predominant meaning of the word until the 9th legislature. However, from the 10th legislature onwards, its prominence decreases due to competition with sense nodules *justice, investigation* and *corruption in Italian politics*. As in Hu et al. (2019), we find that similar senses join forces and cooperate against others while also competing internally.

cluster: *label*	**Legislatures**																	
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*
hygiene, purity, integrity	100	72	55	70	34	60	33	58	33	36	16	10	8	12	2	4	11	2
justice, investigation	-	-	-	-	-	-	2	1	7	17	36	44	66	18	4	11	17	1
environment, sustainability	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	3	1
environment, ecology	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6
renewable energy	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3
corruption in Italian politics	-	-	-	-	-	-	-	21	8	47	38	10	18	48	20	73	55	10
environment	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3
environment, renewable energy	-	-	-	-	-	-	-	-	-	-	-	-	8	18	2	9	8	5
energy, technology	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6	12
sustainability	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3
word frequency	100	72	55	70	34	60	35	80	48	100	90	64	100	96	28	100	100	46

Table 6.3: Prominence of the word *clean* over time. Additionally, we provide the total frequency of the word over time. A dash indicates that no documents (i.e., 0) are present in that cluster at a specific time.

On average, sense shift values are very low, indicating that sense nodules are enriched with documents that are very similar to those already existing. However, we also notice some exceptional cases with high shift scores, for example, 0.56 and 0.59 for the cluster *justice, investigation* in the time interval 7–8 and 8–9. By examining the prominence values in Table 6.3, we find that these cases are sometimes associated with a very small number of documents (e.g., fewer than 10 documents) rather than indicating a true sense shift, while at other times these values can be attributed to misclassification due to the quality of the considered dataset. The former observation aligns with our previous intuition that computing sense prototypes of large sets of embeddings helps to reduce noise (Periti et al., 2022). Indeed, we observe a negative correlation between sense shift and the number of documents within a given time interval, meaning that the smaller the number of documents in a specific time interval, the more sense shift is affected by noise since the impact of outliers becomes more significant in the process of averaging multiple embeddings (i.e. computing sense prototypes). Thus, we argue that the most significant shifts are related to medium-low sense-shift values. For example, we examined the sentences associated with cluster 0 for legislatures 11 and 12, where a sense shift of 0.11 is predicted. In the 10th legislature, the term *clean* is metaphorically used in the context of honesty, integrity, moral correctness and cleaning up criminality. The presence of comparable sentences in the 11th legislature, with a slightly different connotation emphasizing the removal of corruption, old practices, and dishonesty, suggests a broadening of meaning. For instance, within the 10th legislature, expressions such as "piazza pulita" (clean sweep), "mani pulite" (clean hands), "coscienza pulita" (clean conscience) are present. On the other hand, in the 11th legislature, expressions like "paese pulito" (clean country) and "ambiente pulito" (clean environment) are also present.

Further intriguing results from our analysis of various word and sense nodules are presented in Tables 6.4 and 6.5, respectively.

word	time-interval	polysemy	semantic shift	description
clean (*pulito*)	7–8	2–3	0.15	The term is used in the context of *corruption in Italian politics* in addition to its original associations with *hygiene, purity and integrity*.
violence (*violenza*)	17–18	8–14	0.53	The term is used to encompass not just physical violence, sexual assault, and domestic violence, but also gender-biased violence, indicating a broadening in meaning and context.
abuse (*abuso*)	12–13	1–2	0.00	The term is used in the context of *child abuse* in addition to its original associations with *power abuse*.
abuse (*abuso*)	15–16	2–3	0.15	The term is used in the context of *sexual abuse* in addition to its original associations with *power abuse* and *child abuse*.
climate (*clima*)	11–12	3–3	0.08	The term is mainly used for *environmental and climate issues* in addition to its previous usages for *a type of atmosphere* (e.g., political tension) or *a particular situation* (e.g., festive atmosphere).
woman (*donna*)	8–9	2–3	0.28	In the 9th legislature, the term appears in relation to the bill for the establishment of voluntary military service for women in the *Italian Armed Forces*.
gender (*genere*)	15–16	5–6	0.08	The term has evolved beyond its original usage as a means to denote a *kind* or *type* of something and has acquired a new connotation related to *gender identity* and *sexual gender*.
seizure (*sequestro*)	5–6	1–2	0.03	The term underwent a semantic shift, expanding from its original meaning of *seizure* to also refer to the act of *person kidnapping*, due to the first kidnapping for extortion on December 18, 1972.

Table 6.4: Example of semantic shift associated with the corresponding word, time interval, polysemy, and a short description.

6.5 Evaluation on reference benchmarks

As a final test for assessing the effectiveness of WiDiD on LSC, we considered the evaluation framework defined at SemEval-2020 (Schlechtweg et al., 2020). Specifically, we rely on two of the LSC tasks presented in Chapter 2:

word	label	time-interval	prominence	sense shift	description
clean (*pulito*)	hygiene, purity, integrity	7–8	16–10	0.11	The sense nodule has undergone a “broadening” shift. In the 7th legislature, it was related to concepts like *honesty, moral correctness, fighting criminality*. In the 8th legislature, its scope expanded to include *eliminating deception and pollution*, and *cleaning up the old regime*. In the 8th legislature, expressions like *clean sweep*, *clean country*, and *clean environment* emerge. This shift can be attributed to investigations such as “The Mani Pulite” and “Tangentopoli” scandals that revealed a fraudulent and corrupt system.
environment (*ambiente*)	environmental administration; environmental management; environmental protection	8–9	100–100	0.15	The sense nodule exhibited a“broadening” shift. In the 8th legislature, it was related to concepts like *political environment, work environment*. In the 9th legislature its scope expanded to include *ministerial issues* and *environmental bodies* for environmental protection. This shift can be attributed to the establishment of the Ministry of the Environment during the 9th legislature.
right (*diritto*)	law, human right; international right	7–8	26–33	0.17	The sense nodule exhibited a broadening shift. During the 7th legislature, it was primarily associated with concepts such as *law, legal norms*, and *human rights*. In the 8th legislature, its scope expanded specifically in relation to *human rights*. This shift can be attributed to the international agreement known as the Vienna Convention on the Law of Treaties. Indeed, expressions like *Vienna Convention* and *international law* emerged during the 7th legislature, while in the 8th legislature, expressions like *right of* emerged.
party (*partito*)	political parties; Left parties	11–12	96–97	0.11	The sense nodule exhibited a shift in meaning. During the 11th legislature, it was primarily associated with concepts such as *Left parties*, *political party*, and *transparency*. In the 12th legislature, its contextual scope expanded to include the idea of *coalition*. This shift can be attributed to the birth of the Italian People’s Party. Terms like *Socialist Party* and *Democratic Party* emerged in the 8th legislature, while the 12th legislature witnessed the emergence of the expression *Italian People’s Party*.
violence (*violenza*)	violence in social contexts	12–13	28–48	0.21	The sense nodules shifted, expanding from *physical violence* in the 12th legislature to also include *sexual assault* in the 13th legislature.
opposition (*opposizione*)	social opposition; political opposition	8–9	48–34	0.15	The sense nodule exhibited a narrowing shift in meaning. In the 8th legislature, it primarily pertained to the concept of *political opposition*. In the 9th legislature, its contextual expansion included a specific emphasis on *the role of political opposition* and *its significance as a critical* voice.
abortion (*aborto*)	numerical incidence and social implications of abortion	16–17	13–16	0.20	The sense nodule exhibited a narrowing shift, a shift in focus. In the 16th legislature, it was primarily associated with concepts such as *forced, illegal, and clandestine abortions*, as well as *women’s healthcare*. During the 17th legislature, attention turned towards concern regarding the *rising number of medical staff who were conscientious objectors to providing* abortion and its potential impact on *increasing forced, illegal, and clandestine abortions*.

Table 6.5: Example of sense shift associated with the corresponding word, time interval, prominence and a short description.

1. **Binary Change Detection -** ***binary classification*** (Subtask 1): *For a set of target words, decide which words lost or gained usage(s) between C1 and C2, and which did not.* A binary label ($l \in \{0, 1\}$) is assigned to each target word via manual annotation. Then the semantic change word classification computed by a model is evaluated by the Accuracy over the human-annotated test data.

2. **Graded Change Detection -** ***ranking*** (Subtask 2): *Rank a set of target words according to their degree of semantic change between C1 and C2.* A continuous score is assigned to each target word via manual annotation. Then the semantic change word ranking computed by a model is evaluated by Spearman’s rank-order correlation over the human-annotated test data.

We evaluate WiDiD on seven benchmarks that contain a textual diachronic corpus in a given language and test-set of target words, where each word is associated with a change score derived by manual annotation. Table 6.6 summarizes the benchmarks considered. It is worth noting that the evaluation for DIACRIta was executed only on Subtask 1, since no continuous labels are provided. Conversely, the evaluation for RuShiftEval2021 was executed only on Subtask 2, since no binary labels are provided. Furthermore, the Russian corpus of RuShiftEval2021 spans three historical periods, allowing a further demonstration of WiDiD’s effectiveness and robustness in detecting semantic change over time. Note that no benchmarks are currently available over more than two multiple, consecutive time intervals.

		Periods	Tokens	Reference	Target Words
SemEval					
English	C_1	1810–1860	6 M	(Schlechtweg et al., 2020)	37
	C_2	1960–2010	6 M		
Latin	C_1	-200–0	65 k	(Schlechtweg et al., 2020)	40
	C_2	0–2000	253 k		
German	C_1	1800–1899	70.2 M	(Schlechtweg et al., 2020)	48
	C_2	1946–1990	72.3 M		
Swedish	C_1	1790–1830	71.0 M	(Schlechtweg et al., 2020)	31
	C_2	1895–1903	110.0 M		
DIACRIta					
Italian	C_1	1945–1970	52 M	(Basile et al., 2020)	18
	C_2	1990–2014	196 M		
RuShiftEval					
Russian	C_1	1700–1916	94 M	(Kutuzov and Pivovarova, 2021b)	99
	C_2	1918–1990	123 M		
	C_3	1992–2016	107 M		
LSCDiscovery					
Spanish	C_1	1810–1906	13.0 M	(Zamora-Reina et al., 2022b)	100
	C_2	1994–2020	22.0 M		

Table 6.6: Period, size in tokens, reference, and number of target words for the evaluation benchmark considered.

6.5.1 Preliminary results

In this section, we present the results obtained from a preliminary evaluation of WiDiD focusing solely on Subtask 2. Our preliminary evaluation was conducted using the English and Latin corpora of SemEval-2020. The goal of this evaluation was to compare the use of APP within WiDiD with the use of AP and IAPNA clustering, as well as comparing contextualized BERT embeddings with pseudo-contextualized Doc2Vec embeddings. BERT-like models generate dynamic embeddings for a word based on their contextual sequences, whereas Doc2Vec (Le and Mikolov, 2014) produces a static lookup table of word and sequence embeddings only for words and sequences seen during training. We leverage Doc2Vec by computing *pseudo*-contextual word embeddings under the assumption that word occurrences within similar sequences share the same meaning. This implies that, given a target word w in the corpus C_j, we consider Φ_w^j as the set of sequence embeddings related to sequences where w occurs. For training Doc2Vec models, we utilize the Gensim library (Rehurek and Sojka, 2011). Specifically, we train word and sequence embeddings of size 100 for 15 epochs, with a window size of 10. As for BERT, we use a specific model for each language, namely *bert-base-uncased* for English and *bert-base-multilingual-uncased* for Latin.

For the sake of comparison, we also test various evaluation metrics presented in Chapter 2, namely JSD, PDIS, and PDIV, applied to the clusters of contextual embeddings obtained by using AP, IAPNA, and APP, respectively. Since PDIS and PDIV are extensions of the CD and DIV measures, we consider them as additional baselines.

However, in this preliminary evaluation, we only consider instances of the lemma form of target words. This means that we did not perform lemmatization to capture the different occurrences of a target word in various forms (e.g., plural, singular).

			Latin (Spearman's coefficients)			**English (Spearman's coefficients)**		
Clustering	**Training**	**Model**	*JSD*	*PDIS*	*PDIV*	*JSD*	*PDIS*	*PDIV*
AP	trained	Doc2Vec	**0.485***	0.229	-0.023	**0.514***	0.139	0.134
	pre-trained	BERT	**0.394***	0.347*	0.236	0.356*	0.326*	**0.406***
IAPNA	trained	Doc2Vec	**0.462***	0.354*	-0.005	0.199	0.322*	**0.336***
	pre-trained	BERT	**0.411***	0.356*	-0.148	0.336*	**0.499***	0.213
APP	trained	Doc2Vec	$\mathbf{0.512_0}$*	0.337_0*	0.328_0*	$\mathbf{0.333_0}$*	0.077_0	-0.078_0
	pre-trained	BERT	$\mathbf{0.361_0}$*	0.210_0	0.036_0	0.302_0°	$\mathbf{0.512_5}$*	0.370_5*
			CD	*DIV*		*CD*	*DIV*	
	trained	Doc2Vec	0.258°	0.138	-	0.092	0.010	-
	pre-trained	BERT	0.306*	-0.017	-	0.486*	0.168	-

Table 6.7: Spearman's correlation coefficients over different setups with Latin and English corpora. The asterisks denote statistically significant correlations ($p \leq 0.05$), while degree symbols denote low-level correlations with ($0.05 \leq p \leq 0.1$). The subscript index indicates the value adopted for the aging index. We report in bold the highest scores for each clustering-based method considering BERT and Doc2Vec.

Preliminary results of our evaluation are shown in Table 6.7. Surprisingly, Doc2Vec proved to be a suitable model for LSC, in both incremental and non-incremental clustering contexts. It performs well, while being smaller and faster than contextualized models. In particular, Doc2Vec-based methods achieve the highest result in our experiments on both Latin and English, with correlation coefficient of .512 and .514, respectively. APP provides top results on both Latin and English, although AP has a slightly higher performance on English.

On average, both incremental clustering algorithms IAPNA and APP perform well in LSC compared to the conventional AP clustering. We note that IAPNA and APP have opposite behavior on Latin and English: IAPNA has higher results with BERT embeddings on Latin and Doc2Vec embeddings on English, while APP has higher results with Doc2Vec embeddings on Latin and BERT embeddings on English, respectively. The fact that IAPNA and APP perform differently on different languages is consistent with the literature results (Kutuzov and Giulianelli, 2020).

As a further remark, we note that APP produces a smaller and more reasonable number of clusters compared to both AP and IAPNA. For instance, we observed situations where both AP and IAPNA produce more than 100 clusters, which is rather unrealistic if we assume that a cluster represents a word meaning. On the opposite, in our experiments, the number of APP clusters generally varies between 0 and 30. We also note that APP is sensitive to the aging index. In Table 6.7, we present the top results obtained with two different values of the aging index (i.e., 0 and 5). Removing clusters containing less than 5% of the embeddings has a positive impact just in some experiments with English, but not with Latin. We plan to further investigate the effects of the aging index in our future work.

About the measures for LSC, we note that they always perform better than the baselines CD and DIV. We also note that the CD baseline does not work well on Doc2Vec embeddings, while DIV does not work well in all our experiments. On Latin, the highest results are achieved by JSD on both Doc2Vec and BERT embeddings. On English, the top JSD and PDIS results are on Doc2Vec and BERT embeddings, respectively.

More experiments are required on PDIV since it performs very differently in the various experiments we performed, and it achieves statistical significance only in four out of twelve experiments (six on Latin, six on English).

All in all, we note that both IAPNA and APP are competitive when compared to the considered literature approaches.

6.5.2 Detailed results

In this section, we present further results obtained from a detailed evaluation of WiDiD focusing on both Subtask 1 and Subtask 2. This evaluation was conducted on seven benchmarks by considering all the possible forms in which the target words appear.

We used a monolingual BERT model for each language, namely *bert-base-uncased* for English, *simple-latin-bert-uncased* for Latin, *bert-base-german-cased* for German, *bert-base-swedish-cased* for Swedish, *bert-base-spanish-wwm-uncased* for Spanish, *bert-base-italian-cased* for Italian, and *rubert-base-cased* for Russian. The models are base versions of BERT with 12 attention layers and 12 hidden layers of size 768. Furthermore, we compared the use of BERT models with two different multilingual models, both with 12 attention layers and 12 hidden layers of size 768, that is, mBERT *bert-base-multilingual-cased* and XLM-R *xlm-roberta-base*.

Furthermore, going with the intuition that sense prototypes can be beneficial in limiting noise in the vector representations, we compared the use of JSD with the measure based on sense nodules proposed by Kashleva et al. (2022). Following Kashleva et al. (2022), we define the semantic change $\mathcal{S}_w$ as the average pairwise distance (APDP) between all pairs of the sense prototypes $\mu^t_{w,1..k} \in M^t_w$ and $\mu^{t-1}_{w,1..k} \in M^{t-1}_w$. Intuitively, the higher $\mathcal{S}_w$, the more the word w has changed in meaning. This decision stemmed from empirical results in our initial experiments, which consistently demonstrated the superiority of using the canberra distance over the cosine distance.

In line with previous work, for Subtask 1, we binarized the score of a word by using the threshold θ that maximizes the overall result on the test set. Intuitively, the label 0 is assigned to a word if its JSD/APDP score is lower than θ, otherwise the label 1 is assigned to the word. It is worth noting that, development and training sets are not available for the majority of the benchmark, as LSC is typically framed in an unsupervised scenario (Schlechtweg et al., 2020). Therefore, the evaluation of Subtask 1 only provides an indication of the models' capability to recognize semantic change. Indeed, the threshold is set based on the test set. This is also the reason why Subtask 2 is far more popular than Subtask 1 (Periti and Montanelli, 2024). For Subtask 2, we directly used the JSD and APDP scores as the degree of semantic change.

For the sake of comparison, we report the top state-of-the-art results achieved using contextualized embeddings for Subtask 1 and Subtask 2 in Table 6.9 and Table 6.8, respectively. To ensure a fair comparison, we exclusively report results obtained by unsupervised approaches leveraging contextualized embeddings. In addition, it is worth noting that we are reporting the best result achieved in multiple experiments (e.g., using different models and measures). Accordingly, we have compared our best results with the provided state-of-the-art results.

	SemEval				DiacrIta
References	*English* *C*1 - *C*2	*Latin* *C*1 - *C*2	*German* *C*1 - *C*2	*Swedish* *C*1 - *C*2	*Italian* *C*1 - *C*2
Unsupervised					
Kanjirangat et al., 2020	.541	.375	.708	.742	-
Martinc et al., 2020c	.703*	.700	.667*	.710*	-
Karnysheva and Schwarz, 2020	.568	.650	.583	.645	-
Rother et al., 2020	.622	.575	.729	.742	-
Cuba Gyllensten et al., 2020	.568	.675	.562	.710	-
Wang et al., 2020	-	-	-	-	.610*
Giulianelli et al., 2022	.459*	.500*	.521*	-.516*	.389*
Supervised					
Ma et al., 2024a	.784	.700	.813	.806	-
WiDiD	**.757**	**.750**	**.729**	**.774**	**.944**

Table 6.8: Subtask 1: accuracy scores achieved from various state-of-the-art experiments. Asterisks denote scores obtained via fine-tuning contextualized models, while hyphens indicate unavailable experimental results. Bold denotes the best unsupervised scores.

	SemEval				LSCDiscovery	RuShiftEval		
References	*English* *C*1 - *C*2	*Latin* *C*1 - *C*2	*German* *C*1 - *C*2	*Swedish* *C*1 - *C*2	*Spanish* *C*1 - *C*2	*Russian* *C*1 - *C*2	*Russian* *C*2 - *C*3	*Russian* *C*1-*C*3
Unsupervised								
Kanjirangat et al., 2020	.159	.231	.525	.141	-	-	-	-
Martinc et al., 2020c	.436*	.481	.528*	.238*	-	-	-	-
Karnysheva and Schwarz, 2020	.155	.177	.388	.062	-	-	-	-
Rother et al., 2020	.306	.321	.605	.268	-	-	-	-
Cuba Gyllensten et al., 2020	.209	.399	**.656**	.234	-	-	-	-
Montariol et al., 2021	.456*	**.488***	.561*	**.561***	-	-	-	-
Giulianelli et al., 2022	.127*	.318*	.287*	-.108*	-	.247*	.267*	.362*
Kashleva et al., 2022	-	-	-	-	**.553***	-	-	-
Supervised								
Aida and Bollegala, 2024	.774	.124	.902	.656	-	.805	.811	.846
Cassotti et al., 2023a	.757	-.056	.877	.754	-	.799	.833	.842
WiDiD	**.651**	.433	.527	.499	.544	**.273**	**.393**	**.407**

Table 6.9: Subtask 2: Spearman's correlation coefficients achieved from various state-of-the-art experiments. Asterisks denote scores obtained via fine-tuning contextualized models, while hyphens indicate unavailable experimental results. Bold denotes the best unsupervised scores.

Table 6.10 presents the results of our evaluation for both Subtask 1 and 2. For Subtask 1, we note that our results have the potential to outperform the results shown in Table 6.8 across all evaluated benchmarks. Specifically, for the DIACRIta benchmark, which is relevant for our study due to the shared language of our case study corpus, both BERT+JSD and mBERT+JSD exhibit equal effectiveness by correctly labeling 17 out of 18 words. For Subtask 2, our results outperform state-of-the-art results for English and Russian, while being comparable with the state-of-the-art results for the other benchmarks.

As a general remark, and in line with the finding of Kutuzov and Giulianelli (2020), we note that the measure that produces a more uniform predicted score distribution (APDP) works better for the test sets with skewed gold distributions, and the measure that produces a more skewed predicted score distribution (JSD) works better for the uniformly distributed test sets.

As for the model comparison, we observed that, on average, different models achieve similar results for Subtask 1. However, the selection of the model is crucial for Subtask 2. For instance, both BERT and XLM-R demonstrate good performance for English, while the use of mBERT leads to significantly worse results. Interestingly, contrary to the widespread belief that monolingual models are more suitable than multilingual ones, we found that only for English (Subtask 2) and Spanish (Subtask 1 and 2) did employing a monolingual BERT model prove more effective than using a multilingual model. Additionally, despite the expectation that XLM-R would outperform mBERT due to the larger amount of training data and parameters it uses, we observed that mBERT is the most suitable model for Latin (Subtask 1) and Russian (Subtask 2).

		SemEval				**LSCDiscovery**	**RuShiftEval**			**DiacrIta**
	JSD / APDP	*English* *C*1 - *C*2	*Latin* *C*1 - *C*2	*German* *C*1 - *C*2	*Swedish* *C*1 - *C*2	*Spanish* *C*1 - *C*2	*Russian* *C*1 - *C*2	*Russian* *C*2 - *C*3	*Russian* *C*1-*C*3	*Italian* *C*1 - *C*2
Acc Sub. 1	BERT	.622 / .730	.675 / .625	**.729** / .708	.742 / **.774**	**.688** / **.688**	-	-	-	**.944** / .833
	mBERT	.649 / .676	**.750** / .675	**.729** / .646	.742 / **.774**	.675 / .638	-	-	-	**.944** / .722
	XLM-R	.622 / **.757**	.725 / .650	**.729** / .708	**.774** / **.774**	.675 / .625	-	-	-	.889 / .833
Corr Sub. 2	BERT	.256 / **.651**	.334 / .165	.407 / .363	.012 / .155	.429 / **.544**	.198 / .204	.265 / .238	.271 / .177	-
	mBERT	.244 / .237	.410 / -.093	.397 / .280	.015 / .132	.450 / .420	.263 / **.273**	.348 / **.393**	.398 / **.407**	-
	XLM-R	.291 / .635	**.433** / -.096	.225 / **.527**	.087 / **.499**	.463 / .322	.021 / .132	.328 / .250	.292 / .256	-

Table 6.10: Evaluation scores for Subtask 1 and Subtask 2 achieved via accuracy (Acc) and Spearman's correlation coefficients (Corr), respectively, over different benchmarks and setups. For each benchmark, we report our results obtained by using different contextualized models (i.e, BERT, mBERT, XLM-R) and different semantic shift measures (i.e., **JSD / APDP**). We report in bold the highest scores for each benchmark and subtask.

6.6 Discussion and considerations

Data quality. One crucial aspect of diachronic corpora is that the number of documents is often imbalanced, and the presence of a target word is not equally reflected in all the time points considered. In common scenarios, more documents are available for more recent time periods and *it may not be possible to achieve balance in the sense expected from a modern corpus* (Tahmasebi and Dubossarsky, 2023). Furthermore, the quality of the analyzed data can significantly influence the results. Similar to the imbalance issue, the quality of the data is generally higher for recent documents than for past documents. Old documents are often digitized as images using an OCR scanning process to convert them into text. However, this procedure can introduce *OCR errors* that contribute to degrading the quality of the analysis.

In our application of WiDiD, the imbalance was also caused by the inherent varying duration of papacies and legislatures in addition to the availability of documents. For example, a legislature is usually associated with a time period of up to 5 years, which corresponds to the duration of an election cycle. However, in cases where the Parliament withdraws its support from the government through a *vote of no confidence*, the duration can be shorter.

In terms of data quality, the documents in our Parliament corpus were originally stored as images and digitized through an OCR scanning process. As a result, several characters were misrecognized, omitted, or erroneously inserted, distorting the original text across all the legislatures. Although a precise estimation of the extent of these errors is currently unavailable, we enforced heuristics to mitigate OCR errors and retain only the highest-quality sentences in the corpus. Despite the efforts to remove highly corrupted sentences, some errors persist and the processing has further increased the existing imbalance in the corpus.

These issues affect the quality of contextualized embeddings generated by BERT-like models. Thus far, only a few studies have explored the influence of OCR errors on contextualized embeddings (Todorov and Colavizza, 2022; Jiang et al., 2021). As a result, the impact of OCR errors on contextualization remains unclear, and quantifying their effect is challenging. Nevertheless, we hypothesize that there might be significant side effects. For instance, one common problem caused by OCR errors is the inconsistent use of punctuation, resulting in longer or shorter sentences that degrade the quality of the embeddings. Additionally, OCR often introduces or removes spaces, which disrupts sentence segmentation. For example, the word `aperitivo` (i.e., *happy hour*) may become a three-word expression like `ape re timo` (in English, *bee king thyme*), thus affecting the correct interpretation of the sentence. The meaning of words can be also altered by OCR errors that remove accents. For instance, `papa` and `papà` have different meanings (*pope* and *father*, respectively).

In a study on diachronic word sense discrimination (Tahmasebi et al., 2013), the authors showed that due to the design of the algorithm, the quality of the clusters did not degrade with decreasing quality of the corpus, but the number of clusters was radically reduced. When using contextualized embeddings this is not the case, since we can produce embeddings for each occurrence of a target word regardless of the quality of the sentence. As long as the word we are interested in is correctly spelled, its contextual representation will contribute to the meaning of the word, however, with reduced quality. Thus, with contextualized embeddings, the quality of the output inherently depends on the quality of the input data. Due to the significant number of OCR errors in our case study, our empirical results may be less accurate and reliable. However, we expect the OCR errors to affect the corpus at each time period roughly evenly, and thus all senses of a word should be affected to the same degree in any given time period. As a result, small clusters may not be detected and some clusters could show up later than expected. Nevertheless, the case study serves its purpose in demonstrating the functionality of WiDiD but **is not meant as an in-depth, exploratory social science or linguistics study of the Italian parliament**.

There are limitations that must be considered in the context of this case study. Specifically, we predefined a set of target words for analysis without applying the WiDiD approach to the entire vocabulary. Since this case study focuses on a specific domain, it potentially limits the contexts in which some of the targets are typically used. Furthermore, limitations also arise when working with language models such as BERT, which may be trained on a corpus that differs significantly in topics and time periods from our domain.

In this work, we have provided a link to the original website from which our data was collected, as well as a repository link where the dataset used in our study can be accessed. However, we have chosen not to release the complete dataset in its current form. As discussed, the complete dataset contains a significant number of spurious characters and OCR errors, and we are currently undertaking an extensive post-OCR cleaning process to ensure its accuracy for future release, along with comprehensive analytical insights. This cleaning process poses considerable challenges, even with the assistance of advanced generative language models. While these models can aid in correcting OCR errors, they tend to paraphrase or creatively reconstruct sentences (Boros et al., 2024), potentially introducing artifacts that could affect the analysis of lexical semantic changes and the overall reliability of our historical, societal, and political corpus.

Incremental LSC. Incremental LSC enables a more fine-grained analysis of semantic change by tracing the evolution of different word meanings over time. However, semantic change is not uniform across all words or domains. Some words may experience rapid changes in meaning, while others can change gradually or remain relatively stable. Therefore, computational approaches to LSC need to be flexible enough to handle both short- and long-term semantic changes. In addition, word meanings do not necessarily change in a linear way. They are not strictly limited to increasing, decreasing, or remaining stable in prominence. Instead, word meanings can be influenced by various circumstances, leading to both regular and irregular trends that can activate or deactivate meanings in different time periods. These properties make a complete modeling of semantic change extremely complex. While we are advancing existing state-of-the-art change detection methods significantly, we have reduced the complexity in several ways and made several design choices that can affect the results. We discuss a few of these choices below.

First, we chose not to perform online clustering of elements (i.e., sentences with a target word) one-by-one but instead to consider all elements stemming from a time period at the same time. Conducting the clustering step of WiDiD after adding a single new element would enforce clustering on a small number of elements, namely the newly added element and the previous *n* sense prototypes. Such a procedure, which does not correspond to our typical research scenario, is unlikely to result in converging clusters and can lead to erroneously merged clusters, thus losing the"memory" already gathered. We thus opted to cluster all elements from a time period together with the previous sense prototypes all at once, leading to more robust clustering results. While this procedure increases the overall amount of data while clustering, it does not handle gradual semantic change, where only a few elements of a new cluster may initially be present. Consequently, recognition of a semantic change is likely to occur at a later stage, when a consistent amount of evidence supporting the change is considered. To overcome this issue, an approach that combines WiDiD with global evolutionary clustering can be considered. Specifically, if the evidence for establishing a new sense is insufficient within a specific time period, WiDiD will misclassify it. However, because of the *What is done is done*-paradigm, an assignment will never be reconsidered even if additional evidence becomes available in later time periods. This means that, in order to recognize a new sense, the evidence for that sense must be substantial within a specific time period, rather than cumulative across all processed periods. A similar issue may occur when evidence for the establishment of a new sense is sufficient within a certain time period, but some word occurrences denoting the new sense are incorrectly associated by WiDiD with another active sense. This misclassification can lead to a downsample of evidence for the new sense, causing it to be underrepresented and not recognized until more supporting evidence becomes available in later time periods. Thus, the iteration frequency of WiDiD, along with the characteristics of the data under analysis must be carefully considered, taking into account both the risk of disambiguation errors and the possibility of overlooking emerging senses. To overcome this issue, an approach that combines WiDiD with global evolutionary clustering can be considered to review previous assignments and potentially reverse them as necessary.

In WiDiD each sense nodule is currently represented by a single-sense prototype representation, with the same importance as a new element (i.e., contextualized embedding of a word). This approach leads to a higher risk of sense nodules being merged or confused over time. Empirical results indicate that while some clusters persist over time even without the integration of new elements, the majority tend to merge with other clusters over time. In the final step this results in an increase in the number of clusters stemming from the last time period and a decrease in the number of clusters stemming from earlier periods (since in the earlier time periods there were more opportunities for merging). While the aggregation of sense nodules may sometimes aid in focusing on lexicographic meaning (rather than just on sense nodules), at other times it results only in noise representations. This problem could possibly be solved by using a different weighting schema for sense nodules and new elements, but manually annotated ground truth data is needed to perform large-scale evaluation so as to choose the best weighting schema.

Moreover, WiDiD currently considers all occurrences of a word without additional pre- and post-processing. Additional processing techniques could be employed to initially discard ambiguous word occurrences (e.g., where the context is too limited to understand the meaning), or to refine the memory of active meanings at the end of each Incremental Clustering step. For instance, applying a threshold over cluster integrations can distinguish between valid updates (e.g., active clusters enriched with at least *n* elements) and invalid updates (e.g., active clusters enriched with fewer than *n* elements), which should be discarded. A similar threshold can also be applied to cluster merging. Yet another threshold can be employed to classify sense clusters as "lost" or no longer active. Specifically, each cluster can be associated with an aging index to measure how recently it has been updated during incremental clustering, with the threshold determining when it should be considered lost and removed from memory (Castano et al., 2024; Periti et al., 2022). Nevertheless, implementing such thresholds requires careful consideration of the data (e.g., size, domain, time periods, style) and the nature of semantic change under analysis. For example, in studies with limited or high-quality data, a cluster integration of one or a few elements might be a valid update, whereas in studies with extensive or medium-quality data, such minor updates could be considered noisy and disregarded. Similarly, in scenarios where the focus is on detecting immediate changes, such as in rapidly evolving fields, a few intervals without cluster integrations may suffice to deem a sense cluster as lost; conversely, when the focus is on periodic senses a few intervals may not suffice and prematurely pruning those senses from the memory could lead to the undesirable detection of change each time they appear and disappear from memory.

When it comes to interpreting semantic change across multiple time points, two different approaches can be adopted: a evolutionary analysis (first application) and a posteriori analysis (second application). In a posteriori analysis, the snapshot associated with the clustering result of the last iteration is used. Thus, the cluster membership distribution across different time points is considered with respect to the clustering result of the final iteration. That is, we do not consider two clusters individually in previous time periods if they have been merged by the last time period. This analysis focuses on examining how the clusters are distributed and assigned across time, providing insights into the temporal patterns of semantic change and is a simplification of the full LSC problem. Evolutionary analysis, on the other hand, emphasizes the behavior of the clusters themselves rather than their specific distribution across time. It investigates the evolution of clusters, such as their merging or integration over time. Observing changes in cluster composition and structure can yield valuable information regarding the dynamic nature of semantic change (Hu et al., 2019).

In our applications, we have prioritized a posteriori analysis over evolutionary analysis. We chose not to implement any processing thresholds in our WiDiD application, as it was convenient for illustrating the applicability of WiDiD and the complete history of each cluster during the considered time periods. We are currently working on developing more advanced measures and techniques to present the patterns captured by *evolutionary analysis* (i.e., incremental analysis of new sense nodules, their merging and integration), with the aim of constructing a diachronic and hierarchical sense inventory. However, such analysis requires large-scale evaluation across multiple time points and is significantly more complex (see Chapter 4). To be a useful research tool, evolutionary analysis also requires ways to represent the results without overloading the user. We are currently working on creating evaluation data for such a scenario.

Finally, recent research has demonstrated that embeddings lie in an anisotropic space, indicating that all vectors are within a narrow cone. The consequence is that even embeddings of unrelated words may be close together in distributional space and thus exhibit very high similarity. As a result, if a sense prototype is even slightly distorted, one or more sense prototypes may be incorrectly clustered and the algorithm's results may exhibit a large degree of randomness. A way to overcome this issue might be to project the embeddings onto a larger part of the space (i.e., making the cone wider), thus creating more distance between elements.

Possible Applications of WiDiD. Both historical linguistics and lexicography involve the direct application of LSC. The former compares change patterns across time and languages, and the latter needs to update dictionary entries on the basis of new information from modern or historical texts. Much of this work requires manually labeling and interpreting each cluster, which can be a time-consuming task, especially when there are large sets of clusters or when many words are considered at once.

We envision a Query Answering system based on WiDiD as a solution to facilitate the interpretation of semantic change and the analysis of specific word meanings over time. WiDiD allows for intelligent filtering, both on the word level and the sense level. For example, one could study particular words in certain periods of time (pre- and post-war, or pre- and post-pandemic are typical periods of study). Alternatively, one could investigate all documents that use a word in a specific sense.

Such fine-grained analysis across temporal dimensions and all senses of a word is an extremely useful tool in research fields where diachronic analysis of word meaning is central. It is, however, important to couple the outcome of an approach like WiDiD with confidence values that reflect the level of certainty associated with an unsupervised model trained on text of varying quality.

Chapter 7

A systematic evaluation of word embeddings

> "*Sometimes when you innovate, you make mistakes. It is best to admit them quickly and get on with improving your other innovations.*"
>
> Steve Jobs

7.1 Introduction

In the previous chapter, we introduced a novel approach to LSC called WiDiD and framed it with respect to the existing literature. As discussed in Chapter 2, the emergence of LLMs has established contextualized embeddings as the preferred tool for addressing LSC tasks (Periti and Montanelli, 2024; Kutuzov et al., 2022b), specifically the task of Graded Change Detection (GCD). Contextualized embedding models differentiate the meanings of words by contextualizing each occurrence with a distinct embedding. However, the generation and processing of contextualized embeddings across entire corpora present scalability challenges in terms of time and memory consumption (Periti et al., 2022; Montariol et al., 2021). Different strategies have been adopted to tackle these challenges, leading to a proliferation of evaluations across diverse settings (e.g., limited samples of benchmarks) and conditions (e.g., pre-trained vs. fine-tuned models). We observed, as a result, that these evaluations on GCD hinder a fair comparison among the performance of different models and approaches.

Moreover, while the GCD task is attracting more and more evaluations, we also observed that it addresses only a partial complexity inherent to the LSC framework established at SemEval-2020 (Schlechtweg et al., 2020). Notably, the framework includes three distinct aspects:

i) **semantic proximity judgments** of word *in-context*;

ii) **word sense induction** based on proximity judgments;

iii) **quantification of semantic change** from induced senses

As a matter of fact, when contextualized embedding models are used to address GCD, cosine similarities among word embeddings serve as surrogate for **(i)**, without evaluation focused on this aspect. Additionally, most approaches to GCD are *form*-based and pass from **(i)** to **(iii)**, sidestepping the intermediate aspect **(ii)**. That is, they quantify semantic change as overall proximity variation, without inducing word senses. Consequently, while these approaches can be evaluated through GCD, they preclude the interpretation of which meaning(s) have changed.

Following Chapter 4, in this chapter, we argue that **(i)** and **(ii)** are equally relevant aspects as **(iii)**, constituting a fundamental aspect of the LSC problem. Their evaluation can provide valuable insights into the current state of LSC modeling, while offering a broader perspective on contextualized embedding models in Natural Language Processing (NLP).[1]

Chapter outline.

This chapter includes materials originally published in the following publication:

> Francesco Periti and Nina Tahmasebi. 2024**a**. A Systematic Comparison of Contextualized Word Embeddings for Lexical Semantic Change. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Pa- pers), pages 4262–4282, Mexico City, Mexico. Association for Computational Linguistics.

In this chapter, we systematically evaluate and compare various models and approaches for GCD under equal settings and conditions. Our evaluation for GCD spans eight different languages. Our results show superior performance of a recent state-of-the-art model called XL-LEXEME (Cassotti et al., 2023a), over various approaches. Additionally, we conduct a novel and comprehensive evaluation of contextualized models, encompassing aspects **(i)** and **(ii)**, by leveraging two well-established tasks in NLP: Word-in-Context (WiC) and Word Sense Induction (WSI). Through this evaluation we assess the efficacy of various models as *computational annotators*.

This chapter is organized as follows. In Section 7.2, we provide background information on benchmark construction for LSC while also discussing issues in existing evaluations. In Section 7.3, we outline the setup established for our evaluation. In Section 7.4, we present a comparison of models and approaches for solving GCD. In Section 7.5, we evaluate contextualized models for WiC, WSI, and GCD by considering them as annotators. Finally, in Section 7.6, we discuss the implications and limitations of our evaluation.

7.2 Background and related work

The established LSC framework adheres to the novel annotation paradigm for word senses and encompasses **(i-iii)** (Schlechtweg et al., 2021). **(i)** Human annotators provide semantic proximity judgments for pairs of

[1] https://github.com/FrancescoPeriti/CSSDetection

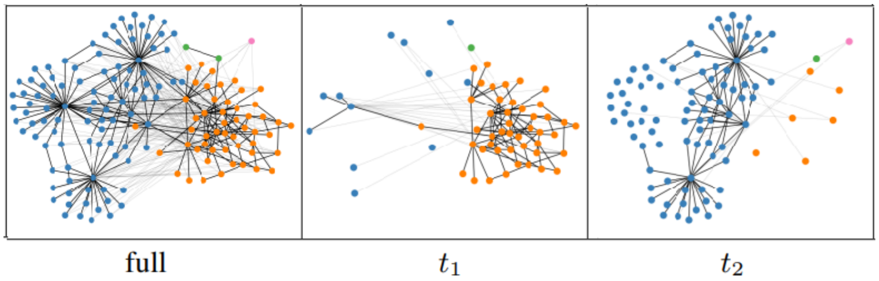


Figure 7.1: DWUG for the German word *Eintagsfliege*. Nodes represent word usages. Edges represent the relatedness between usages. Colors indicate clusters (senses) inferred from the full graph (Laicher et al., 2021).

word usages *sampled* from a diachronic corpus spanning two time periods. **(ii)** Word usages and judgments are represented as nodes and edges in a weighted, *diachronic* graph, known as Diachronic Word Usage Graph (DWUG). This graph is then clustered with a graph clustering algorithm and the resulting clusters are interpreted as word senses (see Figure 7.1), thus sidestepping the need for explicit word sense definitions. Finally, **(iii)** given a word, a ground truth score of semantic change is computed by comparing the probability distributions of clusters in different time periods, e.g., a cluster with most of its usages from one time period indicates a substantial semantic change.

Originally, the framework was proposed in a shared task at SemEval-2020, including benchmarks for four languages, namely English (EN), German (DE), Swedish (SV), and Latin (LA) (Schlechtweg et al., 2020). Benchmarks for Italian (Basile et al., 2020), Russian (RU) (Kutuzov and Pivovarova, 2021b), Spanish (ES) (Zamora-Reina et al., 2022b), Norwegian (NO) (Kutuzov et al., 2022a), and Chinese (ZH) (Chen et al., 2023a, 2022a) have recently been introduced. Each benchmark consists of a diachronic corpus and a set of target words over which the human annotation was conducted. The evaluation over a benchmark is typically conducted through the GCD task where the goal is to rank the targets by degree of semantic change across the corpus. The Spearman correlation between *predicted* and *ground truth* scores is used to evaluate models and approaches.

Approaches to Graded Change Detection. As presented in Chapter 2, GCD is typically addressed using two kinds of approaches for modeling word meanings: *form*- and *sense*-based (Periti and Montanelli, 2024; Giulianelli et al., 2020). The former captures signals of change by analyzing how the dominant meaning, or the degree of polysemy of a word, changes over time (e.g., Giulianelli et al., 2020; Martinc et al., 2020a). The latter cluster word usages according to their meanings and then estimate the semantic change of a word by comparing the cluster distribution of its usages over time (e.g., Periti et al., 2024e; Martinc et al., 2020b). Form- and sense-based approaches can be further distinguished into *supervised*, which leverage external knowledge (e.g., dictionaries, Rachinskiy and Arefyev, 2022) or other forms of supervision (e.g., Word-in-Context datasets, Cassotti et al., 2023a), and *unsupervised*, which rely solely on the knowledge encoded in pre-trained models (e.g., Aida and Bollegala, 2023).

Comparison of approaches. Models and approaches for GCD have been evaluated under different settings and conditions. For example, some studies utilized the *entire* diachronic corpus to estimate the change of each target (e.g., Periti et al., 2022), while others relied on smaller *samples* (e.g., Rodina et al., 2021), or solely on the annotated word usages (e.g., Laicher et al., 2021). Also, different versions of the ground truth, each containing a different number of targets, are used (e.g., Schlechtweg et al., 2022a). In the current literature, some studies fine-tune the models on the corpus (e.g., Rosin et al., 2022), while others directly use pre-trained models (e.g., Kudisov and Arefyev, 2022). Performance comparisons are conducted across different models such as BERT (e.g., Laicher et al., 2021), mBERT (e.g., Beck, 2020), and XLM-R (e.g., Giulianelli et al., 2022). However, even when the same model is employed, different layer aggregations are used, such as concatenating the output of the last four encoder layers (e.g., Kanjirangat et al., 2020), or summing the output of all the encoder layers (e.g., Giulianelli et al., 2022). Moreover, sense-based approaches are compared with different clustering algorithms such as Affinity Propagation (e.g., Martinc et al., 2020b), A Posteriori affinity Propagation (e.g., Periti et al., 2022), and K-Means (e.g., Montariol et al., 2021).

As a result, comparing Spearman correlation across different evaluations is often **misleading**.

Current modeling of LSC. Current modeling of LSC overlooks the procedure **(i-iii)** used to generate the ground truth. Mostly, only **(iii)** is evaluated by relying on form-based approaches. However, these approaches capture only the *degree* of semantic change, preventing its interpretation. Sense-based approaches could fill this gap by explaining *how* and *what* has changed, but currently suffer from lower performance on **(iii)** and are therefore less pursued. As a result, it is not clear which meanings these models and approaches are capturing. There is thus a need to carefully evaluate their ability in both **(i)** and **(ii)**.

The evaluation of the shared task participants relied solely on the change values derived from the annotations. In particular, in the shared tasks, the annotated usages were mixed with additional usages to create the training corpora, possibly introducing noise on the derived change scores. The annotated usages were released at a later stage, but they are generally not used for evaluation purposes. To the best of our knowledge, only Laicher et al. (2021) evaluate the aspect **(ii)** through the WSI task by using these annotated usages. This evaluation needs to be extended beyond a single model, using the same procedure used to generate the ground truth. This way, we can comprehensively assess contextualized models by juxtaposing human judgments with embedding similarities, as well as clustering derived from human judgments with clustering derived from embeddings.

A systematic comparison under equal settings and conditions is necessary to evaluate different models and approaches. Thus, we first evaluate standard form- and sense-based approaches to provide a fair performance comparison on GCD across eight languages. We then assess different models as *computational annotators* by evaluating them on **(i-iii)** through WiC, WSI, and GCD. Aligning with Karjus (2023), if computational models perform close to human-level, their usage would represent an unprecedented opportunity to scale up semantic change studies in the humanities and social sciences.

7.3 Evaluation setup

We consider benchmarks for eight different languages: EN, LA, DE, SV, ES, RU, NO, and ZH (see Table 7.1). For each benchmark, we evaluate four different models: BERT (Devlin et al., 2019), mBERT, XLM-R (Conneau et al., 2020), and XL-LEXEME (Cassotti et al., 2023a). Aligning with the *unsupervised* nature of the LSC framework, we compare pre-trained models without performing additional fine-tuning (see Table 7.2). For each model and each target word in a benchmark, we collect contextualized embeddings for all its word usages in both time periods. Specifically, we generate the sets of embeddings $\Phi^1 = \{a_1, ..., a_n\}$ and $\Phi^2 = \{b_1, ..., b_m\}$ for the word usages associated to time periods t_1 and t_2, respectively.

	EN	**LA**	**DE**	**SV**	**ES**	**RU**			**NO**		**ZH**
	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_2 - C_3$	$C_1 - C_3$	$C_1 - C_2$	$C_2 - C_3$	$C_1 - C_2$
Time periods	C_1: 1810 – 1860 C_2: 1960 – 2010	C_1: 200 – 0 C_2: 0 – 2000	C_1: 1800 – 1899 C_2: 1946 – 1990	C_1: 1790 – 1830 C_2: 1895 – 1903	C_1: 1810 – 1906 C_2: 1994 – 2020	C_1: 1700 – 1916 C_2: 1918 – 1990	C_2: 1918 – 1990 C_3: 1992 –2016	C_1: 1700 – 1916 C_3: 1992 –2016	C_1: 1929 –1965 C_2: 1970 – 2013	C_1: 1980 – 1990 C_2: 2012 – 2019	C_1: 1954 – 1978 C_2: 1979 – 2003
Diachronic Corpus	C_1: CCOHA C_2: CCOHA	C_1: LatinISE C_2: LatinISE	C_1: DTA C_2: BZ+ND	C_1: Kubhist C_2: Kubhist	C_1: PG C_2: TED2013, NC MultiUN Europarl	C_1: RNC C_2: RNC C_3: RNC	C_1: RNC C_2: RNC C_3: RNC	C_1: RNC C_2: RNC C_3: RNC	C_1: NBdigital C_2: NBdigital	C_1: NBdigital C_2: NAK	C_1: People's Daily C_2: People's Daily
# targets	46	40	50	44	100	111	111	111	40	40	40
Benchmark version	version 2.0.1 Schlechtweg et al.	version 1 McGillivray et al.	version 2.3.0 Schlechtweg et al.	version 2.0.1 Tahmasebi et al.	version 4.0.0 Zamora-Reina et al.	version 1 Kutuzov and Pivovarova			version 1 Kutuzov et al.		version 1 Chen et al.

Table 7.1: LSC benchmark for Graded Change Detection. Overview of time periods, diachronic corpus composition, number of targets, and benchmark versions used in this study.

	BERT	**mBERT**	**XLM-R**	**XL-LEXEME**
English	*bert-base-uncased*	*bert-base-multilingual-cased*	*xlm-roberta-base*	*xl-lexeme*
Latin	-	*bert-base-multilingual-cased*	*xlm-roberta-base*	*xl-lexeme*
German	*bert-base-german-cased*	*bert-base-multilingual-cased*	*xlm-roberta-base*	*xl-lexeme*
Swedish	*bert-base-swedish-uncased*	*bert-base-multilingual-cased*	*xlm-roberta-base*	*xl-lexeme*
Spanish	*bert-base-spanish-wwm-uncased*	*bert-base-multilingual-cased*	*xlm-roberta-base*	*xl-lexeme*
Russian	*rubert-base-cased*	*bert-base-multilingual-cased*	*xlm-roberta-base*	*xl-lexeme*
Norwegian	*nb-bert-base*	*bert-base-multilingual-cased*	*xlm-roberta-base*	*xl-lexeme*
Chinese	*bert-base-chinese*	*bert-base-multilingual-cased*	*xlm-roberta-base*	*xl-lexeme*

Table 7.2: BERT, mBERT, XLM-R, and XL-LEXEME models employed in our evaluation. All the models are base versions with 12 encoder layers and are available at `huggingface.co`.

Setting 1: standard Graded Change Detection We compare the use of different models with four standard approaches to GCD, specifically two form-based and two sense-based. Similar to Laicher et al. (2021), we consider the raw data originally used to derive ground truth scores, instead of considering the associated corpora. This ensures an accurate evaluation under a controlled setting.

Setting 2: Computational annotators We assess different models as computational annotators by using cosine similarities between embeddings as a surrogate of human judgments. In our evaluation, we consider word usage pairs where human judgments are available, instead of considering all potential usage pairs (as in Setting 1). Specifically, we adhere to the framework **(i-iii)** and evaluate different models through the WiC, WSI, and GCD tasks.

GPT-4 evaluation. Inspired by Laskar et al.; Kocoń et al.; Karjus (2023; 2023; 2023), we evaluate GPT-4 and compare its use to contextualized models. However, the limited accessibility and high associated cost constraint our extension only to the EN benchmark. We evaluate GPT-4 as computational annotator (i.e., Setting 2) by relying on computational proximity judgments gathered through the following method.

We initialized the model with the following *system* prompt (guideline):

```
Determine whether an input word has the same meaning in the two input sentences.
Answer with 'Same', 'Related', 'Linked', or 'Distinct'. This is very important
to my career.
```

Notably, we combine and refine two different prompts used in previous works. We drew inspiration from the prompt utilized by Karjus (2023) to assess GPT-4 in addressing the Graded Change Detection task. Additionally, we drew inspiration from the prompt utilized by Li et al. (2023), called *EmotionPrompt*, which combines the original prompt with emotional stimuli to enhance the performance of LLMs.

For each word usage pair, we used the following *instruction* prompt:

```
Determine whether [Target word] has the same meaning in the following sentences.
Do they refer to roughly the Same, different but closely Related, distant/figu-
ratively Linked or unrelated Distinct word meanings?
Sentence 1: [Context 1]
Sentence 2: [Context 2]
```

Notably, drawing inspiration from the OpenAI documentation[2] and the prompts utilized in previous work for the Word-in-Context task (Periti et al., 2024d; Kocoń et al., 2023; Laskar et al., 2023), we structured our prompt in a format that facilitates parsing and comprehension. For each usage pair $\langle w, c_1, c_2\rangle$ of a word w, we substitute [Target word] with the actual target w and [Context 1] and [Context 2] with c_1 and c_2, respectively.

We prompt GPT-4 without providing any message history. This means that, for each usage pair $\langle w, c_1, c_2\rangle$, we re-initialize the model with the initial prompt (guideline) and subsequently prompt the model to gather a semantic proximity judgment for the pair $\langle w, c_1, c_2\rangle$. This approach ensures that the model relies solely on its pre-trained knowledge, preventing potential biases stemming from previously prompted pairs.

7.4 Comparison of approaches to LSC

We evaluate different approaches for GCD using the Spearman correlation between computational predictions and ground truth scores. Specifically, we process the embeddings of each target using the following

[2] platform.openai.com/docs/guides/prompt-engineering

approaches. We direct the reader to Chapter 2 for further details.

Form-based approaches. In our most recent survey on LSC Periti and Montanelli (2024), we observed that cosine distance over word prototype (PRT) and the average pairwise distance (APD) consistently demonstrated superior performance compared to alternative approaches. Thus, we employ these approaches:

1. **PRT** computes the degree of change of a word w as the cosine distance between the average embeddings μ_1 and μ_2 (also know as *prototype* embeddings) of w in the time periods t_1 and t_2 (Martinc et al., 2020a; Kutuzov and Giulianelli, 2020). Formally, given a word w, we compute its degree of change by computing:
$$\text{PRT}(\Phi^1, \Phi^2) = 1 - cosine(\mu_1, \mu_2) \tag{7.1}$$
The intuition behind PRT is that a prototype embedding encodes the dominant meaning of a word, and as such, the semantic change is computed as a shift in the dominant meaning over time.

2. **APD** computes the degree of change of a word w as the average pairwise distance between the word embeddings in Φ^1 and Φ^2 (Giulianelli et al., 2020; Kutuzov and Giulianelli, 2020). Formally, given a word w, we compute its degree of change, where d is cosine distance, as follows:
$$\text{APD}(\Phi^1, \Phi^2) = \frac{1}{|\Phi^1||\Phi^2|} \cdot \sum_{a \in\, \Phi^1,\ b \in\, \Phi^2} d(a, b) \tag{7.2}$$
The intuition behind APD is that different word embeddings encode the polysemy of a word, and as such, the semantic change is computed as a shift in the word's degree of polysemy.

Sense-based approaches. We choose two state-of-the-art sense-based approaches. The first utilizes the unsupervised clustering algorithm Affinity Propagation (AP) combined with the Jensen Shannon divergence (JSD). Additionally, we employ the evolutionary extension of Affinity Propagation, called A Posteriori affinity Propagation (APP), combined with the average pairwise distances between sense prototypes (APDP). As discussed in the previous chapter, we refer to this approach as WiDiD (Periti et al., 2022).

1. **AP+JSD** leverages the AP clustering to distinguish the different contextual usages of a given word w. Specifically, the embeddings Φ^1, and Φ^2 are *jointly* clustered to generate clusters comprising embeddings from both time periods (i.e., t_1 and t_2), or embeddings exclusive from a time period (i.e., t_1 or t_2). The semantic change of w is computed as the JSD between the probability distributions p_1 and p_2 of clusters in time periods t_1 and t_2. These distributions represent the relative number of embeddings from Φ^1 and Φ^2 grouped in each cluster, respectively (Martinc et al., 2020b,c). Formally, the degree of semantic change is:
$$\text{JSD}(p_1, p_2) = \frac{1}{2}\left(KL(p_1||M) + KL(p_2||M)\right) \tag{7.3}$$

where KL stands for Kullback-Leibler divergence and $M = \frac{(p^1+p^2)}{2}$. The intuition behind AP+JSD is that different clusters encode nuanced word meanings, and as such, the semantic change is computed as an overall measure of the differences in the prominence of each sense over time.

2. **WiDiD** leverages the APP clustering to distinguish the usages of a given word w. Specifically, the embeddings Φ^1, and Φ^2 are *individually* clustered to generate incremental clusters of embeddings that evolve with each clustering iteration. The semantic change of w is computed as the average pairwise distances between the *sense prototypes* Ψ^1 and Ψ^2 of w in the time periods t_1 and t_2, where Ψ^1 and Ψ^2 are the set of embeddings obtained by averaging the embeddings Φ^1 and Φ^2 in each cluster, respectively (Periti et al., 2024e; Kashleva et al., 2022). Formally, given a word w, the degree of semantic change is computed as follows:[3]

$$\text{APDP}(\Phi^1, \Phi^2) = \text{APD}(\Psi^1, \Psi^2) \tag{7.4}$$

The intuition behind WiDiD is similar to AP+JSD. However, while the latter considers change as the difference between the amount of probability for a sense over time, WiDiD is similar to APD in computing the shift in prototypical word meanings.

			EN	**LA**	**DE**	**SV**	**ES**	**RU**			**NO**		**ZH**	**Avg**$_w$
			C_1-C_2	C_1-C_2	C_1-C_2	C_1-C_2	C_1-C_2	C_1-C_2	C_2-C_3	C_1-C_3	C_1-C_2	C_2-C_3	C_1-C_2	C_i-C_j
form-based	APD	BERT	.563	-	.271	.270	.335	.518	.482	.416	.441	.466	.656	.449
		mBERT	.363	.102	.398	.389	.341	.368	.345	.386	.279	.488	.689	.371
		XLM-R	.444	.151	.264	.257	.386	.290	.287	.318	.195	.379	.500	.316
		XL-LEXEME	**.886***	**.231**	**.839***	**.812***	**.665***	**.796***	**.820***	**.863***	**.659**	**.640***	**.731***	**.751***
		SOTA: sup.	*.757*	*-.056*	*.877*	*.754*	*n.a.*	*.799*	*.833*	*.842*	*.757*	*.757*	*n.a.*	
		SOTA: uns.	*.706*	*.443*	*.731*	*.602*	*n.a.*	*.372*	*.480*	*.457*	*.389*	*.387*	*n.a.*	
	PRT	BERT	.457	-	.422	.158	.413	.400	.374	.347	.507	.444	**.712**	.406
		mBERT	.270	.380	.436	.193	.543	.391	.356	.423	.219	.438	.524	.395
		XLM-R	.411	.424	.369	.020	.505	.321	.443	.405	.387	.149	.558	.381
		XL-LEXEME	**.676**	**.506***	**.824**	**.696**	**.632**	**.704**	**.750**	**.727**	**.764***	**.519**	.699	**.693**
		SOTA: sup.	*.531*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	
		SOTA: uns.	*.467*	*.561*	*.755*	*.392*	*n.a.*	*.294*	*313*	*313*	*.378*	*.270*	*n.a.*	
sense-based	AP+JSD	BERT	.289	-	.469	- .090	.225	.069	.279	.094	**.314**	.011	.165	.179
		mBERT	.181	.277	.280	.023	.067	.017	.086	- .116	.035	- .090	.465	.077
		XLM-R	.278	**.398**	.224	-.076	.224	- .068	**.209**	**.130**	- .100	.030	.448	.142
		XL-LEXEME	**.493**	.033	**.499**	**.118**	**.392**	**.106**	.053	.117	.297	**.381**	**.308**	**.223**
		SOTA: sup.	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	
		SOTA: uns.	*.436*	*.481*	*.583*	*.343*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	
	WiDiD	BERT	.385	-	.355	.106	.383	.135	.102	.243	.233	.087	.533	.239
		mBERT	.323	- .039	.312	.195	.343	- .068	.160	.142	.241	.290	.338	.181
		XLM-R	.564	- .064	.499	.129	.459	**.268**	.216	.342	.226	.349	.382	.314
		XL-LEXEME	**.652**	**.236**	**.677**	**.475**	**.522**	.178	**.354**	**.364**	**.561**	**.457**	**.563**	**.422**
		SOTA: sup.	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	
		SOTA: uns.	*.651*	*-.096*	*.527*	*.499*	*.544*	*.273*	*.393*	*.407*	*n.a.*	*n.a.*	*n.a.*	

Table 7.3: Evaluation of standard approaches to GCD in terms of Spearman correlation. Top score for each approach and benchmark in **bold**. The top score of each benchmark is marked with an asterisk (*). We include state-of-the-art performance achieved by *supervised* (sup.) and *unsupervised* (uns.) approaches in *italic*. Avg is the weighted average score based on the number of targets in each benchmark. Results not available denoted as n.a.

[3] Following Periti et al. (2024e), we use the Canberra distance instead of the cosine distance

7.4.1 Evaluation results

Evaluation results- Table 7.3 We present the results of our evaluation in Table 7.3 for both form- and sense-based approaches. For the sake of comparison, we include state-of-the-art (SOTA) results in Table 7.4.[4] As a general remark, we note instances where our results surpass SOTA (e.g., XL-LEXEME+APD for EN). We attribute this to the controlled setting established in our experiments. We note also instances where our results are lower than SOTA (e.g., BERT+APD for SV). This discrepancy may be influenced by various factors such as *different versions* of the benchmarks (e.g., 37 vs 46 targets for EN in DWUG version 2.0.1, Schlechtweg et al., 2020). Additionally, *variations in text pre-processing* can play a beneficial role. For instance, Laicher et al. (2021) demonstrate the effectiveness of lemmatization to mitigate word form biases, while Martinc et al. (2020c) suggest that filtering Named Entities can help models avoid inflating semantic change. Moreover, some studies *fine-tune or utilize different embedding layers*, whereas we adhere to the standard, generally adopted procedures without fine-tuning, considering embeddings generated from the last (i.e., 12th) layer of the models. Finally, there are sometimes significantly different results reported by different studies under similar conditions. For instance, Zhou et al. (2023b) achieve a correlation of .706 using pre-trained BERT and APD, whereas others typically report correlations ranging between .400 and .600 (e.g., .489, Keidar et al., 2022; .514, Giulianelli et al., 2020; .546, Kutuzov and Giulianelli, 2020; .571, Laicher et al., 2021). This disparity cannot currently be explained.

		EN	LA	DE	SV	ES	RU			NO		ZH
		$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_2 - C_3$	$C_1 - C_3$	$C_1 - C_2$	$C_2 - C_3$	$C_1 - C_2$
form-based	**APD**	XL-L. : .757 Cassotti et al. BERT: .706 Zhou et al.	XL-L. : -.056 Cassotti et al. mBERT: .443 Pömsl and Lyapin	XL-L. : .877 Cassotti et al. BERT: .731 Laicher et al.	XL-L. .754 Cassotti et al. BERT: .602 Laicher et al.	n.a. n.a.	XL-L. : .799 Cassotti et al. XLM-R: .372 Giulianelli et al.	XL-L. : .833 Cassotti et al. XLM-R: .480 Giulianelli et al.	XL-L. : .842 Cassotti et al. XLM-R: .457 Giulianelli et al.	XL-L. : .757 Cassotti et al. XLM-R: .389 Giulianelli et al.	XL-L. : .757 Cassotti et al. XLM-R: .387 Giulianelli et al.	n.a. n.a.
	PRT	BERT: .531 Zhou et al. BERT: .467 Rosin et al.	mBERT: .561 Kutuzov and Giulianelli	n.a. BERT: .755 Laicher et al.	n.a. BERT: .392 Zhou and Li	n.a. n.a.	n.a. XLM-R: .294 Giulianelli et al.	n.a. XLM-R: .313 Giulianelli et al.	n.a. XLM-R: .313 Giulianelli et al.	n.a. XLM-R: .378 Giulianelli et al.	n.a. XLM-R: .270 Giulianelli et al.	n.a. n.a.
sense-based	AP+JSD	n.a. BERT: .436 Martinc et al.	n.a. mBERT: .481 Martinc et al.	n.a. BERT: .583 Montariol et al.	n.a. BERT: .343 Martinc et al.	n.a. n.a.	n.a. n.a.	n.a. n.a.	n.a. n.a.	n.a. n.a.	n.a. n.a.	n.a. n.a.
	WiDiD	n.a. BERT: .651 Periti et al.	n.a. XLM-R: - .096 Periti et al.	n.a. XLM-R: .527 Periti et al.	n.a. XLM-R: .499 Periti et al.	n.a. BERT: .544 Periti et al.	n.a. mBERT: .273 Periti et al.	n.a. mBERT: .393 Periti et al.	n.a. mBERT: .407 Periti et al.	n.a. n.a.	n.a. n.a.	n.a. n.a.

Table 7.4: State-of-the-art performance for GCD: Top Spearman correlations obtained across benchmarks by form- and sense-based approaches. For each approach, we report correlation for both *supervised* (above the line) and *unsupervised* (below the line) settings.

Languages. We obtain strong correlations with all benchmarks but LA. Our results show a *weighted average* correlation of **.751** when employing XL-LEXEME + APD. In this calculation, we assign weights based on the number of targets in each benchmark, considering larger sets more reliable than smaller ones. For LA, it can be argued that the models were not directly tailored or fine-tuned for Latin. However, XL-LEXEME demonstrates optimal performance in GCD in SV and medium performance in SP and NO without specific

[4] Our comparison includes results from different benchmarks using the same approaches. However, some benchmarks might have been assessed using other approaches.

training on either. This leads us to consider that the quality of the LA benchmark potentially is lower than other benchmarks, as it was developed using a different procedure (Schlechtweg et al., 2020).

Form-based vs Sense-based. We note that form-based approaches significantly outperform sense-based approaches. Our results consistently highlight APD as the most effective approach, regardless of the skewness in the distribution of judgments, as previously argued by Kutuzov and Giulianelli (2020). In addition, WiDiD consistently demonstrates superior performance over AP+JSD. This can be attributed to the use of i) an evolutionary clustering algorithm, which enables to consider the time dimension of text in a dynamic way; or, alternatively ii) APD over sense-prototypes, as APD has demonstrated high effectiveness.

Our **leaderboard** is as follows: APD, PRT, WiDiD, AP+JSD. Although form-based approaches exhibit superior effectiveness, they fall short in capturing word meanings and interpreting detected semantic changes. In contrast, although sense-based approaches theoretically facilitate such modeling and interpretation, they obtain poor results in GCD, raising concerns about their reliability and whether they capture meaningful patterns or produce noisy aggregation. We will investigate this in Section 7.5.

Supervised vs Unsupervised. We note that the use of supervision significantly improves the modeling of semantic change for both form- and sense-based approaches. While Cassotti et al. (2023a) have previously evaluated XL-LEXEME + APD, we extend the evaluation to sense-based approaches, demonstrating that *supervision* enhances the performance of AP+JSD and WiDiD.

Models. We note that the use of XL-LEXEME significantly improves the modeling of LSC compared to standard BERT, mBERT, and XLM-R. However, we observe a pattern in performance, indicating that on average, BERT performs better than mBERT, which, in turn, performs better than XLM-R for form-based approaches. This suggests that the use of XLM-R models is not more effective than BERT models for LSC, confirming the medium-low correlation coefficients obtained by Giulianelli et al. (2022) using XLM-R.

Layers. As different works employ different embedding layers, we repeat our evaluation by considering embeddings generated by each layer of BERT, mBERT, and XLM-R (see Table B.1). Our evaluation aligns with recent findings on other downstream tasks (Ma et al., 2019; Reif et al., 2019; Liang and Shi, 2023) and shows that using early layers consistently results in higher performance. For example, we note a correlation of .747 for ZH by using layer 4, compared to .656 obtained by using the last layer of BERT. On average, and in line with Periti and Dubossarsky (2023), we find that the best results for each language are obtained by leveraging embeddings from layers 8 – 10.

Furthermore, since previous studies aggregated outputs from different layers, we also use aggregated embeddings extracted from different layers through sum and concatenation. Specifically, our evaluation covers all possible layer combinations with lengths of 2 (e.g., layers 1 and 2), 3 (e.g., layers 6, 7, and 8), and 4 (e.g., layers 9, 10, 11, 12). We find no improvement in aggregating the output of the last four layers for addressing GCD. By employing alternative layer combinations, we obtain a higher correlation compared to both the last layer and the last four layers. For instance, for EN, using the sum of layers 2, 4, 5, and 8 for APD+BERT, or the concatenation of layers 4, 5, 6, and 11 for WiDiD+BERT, results in a correlation of .692 and .760, respectively; compared to .563 (APD) and .385 (WiDiD) by using the last BERT layer (see Appendix B for further results). However, no combination consistently emerges as the optimal choice across various benchmarks or models. Instead, we observe that using a middle layer, such as layer 8, tends to be advantageous across benchmarks and models compared to the last layer or the aggregation of the last four layers (see Figure 7.2 and 7.3).

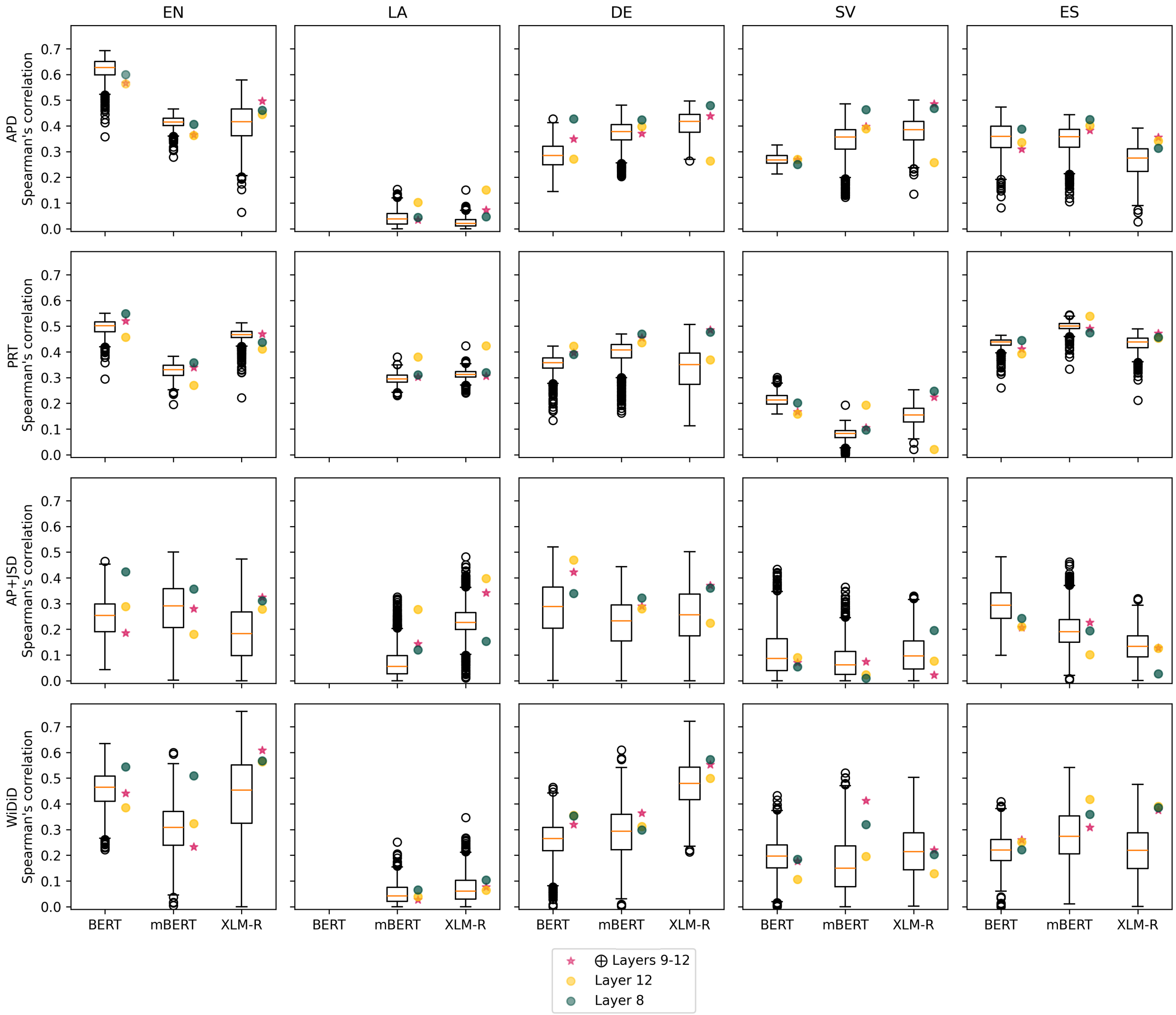


Figure 7.2: Score distribution for GCD obtained by using all possible layer combinations of length 2 (e.g., Layer 1 and 2), length 3 (e.g., Layer 10, 11, 12), and length 4 (e.g., Layer 1, 10, 11, 12) for BERT, mBERT, and XLM-R. The y-axis represents the Spearman correlation. We highlight the performance for GCD obtained using Layer 8, Layer 12, and the sum of the last 4 layers (i.e., $\bigoplus$ 9-12).

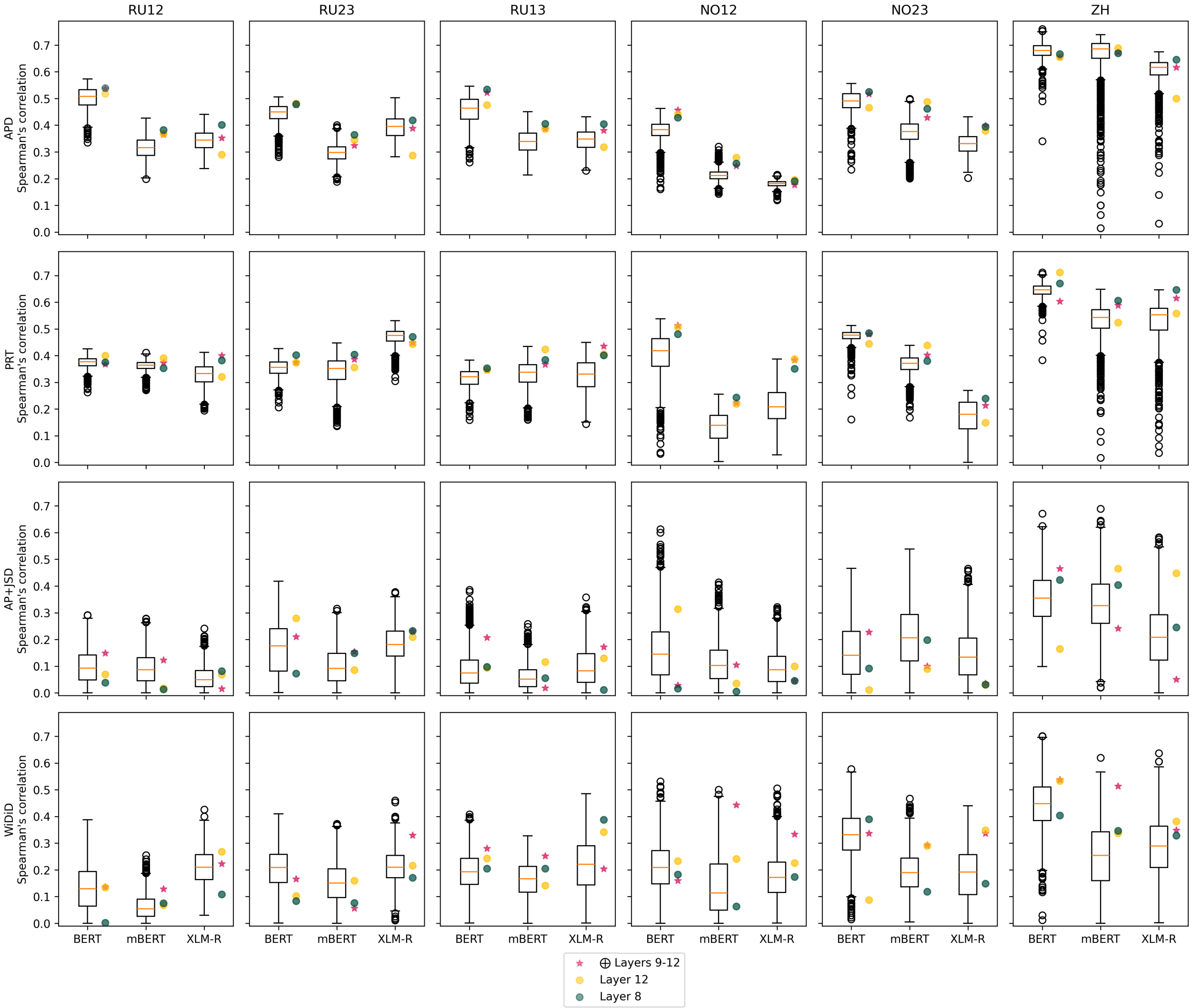


Figure 7.3: Score distribution for GCD obtained by using all possible layer combinations of length 2 (e.g., Layer 1 and 2), length 3 (e.g., Layer 10, 11, 12), and length 4 (e.g., Layer 1, 10, 11, 12) for BERT, mBERT, and XLM-R. The y-axis represents the Spearman correlation. We highlight the performance for GCD obtained using Layer 8, Layer 12, and the sum of the last 4 layers (i.e., $\bigoplus$ 9-12).

7.5 Computational annotation

We evaluate different models on reproducing human judgments **(i)**, the inferred word senses **(ii)**, and the resulting change scores **(iii)**.

We leverage models as annotators, hence the term *computational annotator*, using the same procedure employed for benchmark construction (Schlechtweg, 2023; Schlechtweg et al., 2021, 2020, 2018). However, we cannot evaluate LA as the benchmark was developed differently nor **(ii)** for the RU benchmark since no word senses were provided (Kutuzov and Pivovarova, 2021b).

7.5.1 (i) - Word-in-Context

Given a benchmark, a word usage pair is associated with two contexts, c_1 and c_2, along with the average judgment of multiple annotators. We thus use the cosine similarity between the embeddings of w in the contexts c_1 and c_2 as computational proximity judgment.

Our evaluation is grounded in the Word-in-Context (WiC) task (Loureiro et al., 2022; Raganato et al., 2020; Pilehvar and Camacho-Collados, 2019). In contrast to the original WiC definition, our WiC evaluation aligns with the continuous framework introduced by Armendariz et al. (2020a) in the Graded Word Similarity in Context task. Specifically, we evaluate the quality of computational predictions by computing the Spearman correlation with human judgments.

7.5.2 (ii) - Word Sense Induction

We first create a DWUG using the computational annotations in Section 7.5.1. Then, we derive sense clusters through a variation of correlation clustering (Bansal et al., 2004) on the DWUG.

Our evaluation is grounded in the Word Sense Induction (WSI) task (Aksenova et al., 2022; Manandhar et al., 2010; Agirre and Soroa, 2007). We evaluate the quality of clusters from computationally annotated DWUGs against clusters from human-annotated DWUGs. Specifically, we use Adjusted Rand Index (ARI, Hubert and Arabie, 1985) and Purity (PUR, Manning, 2009) as metrics to quantify the cluster agreement. ARI comprehensively evaluates the similarity among clustering results. However, it may yield low scores when a clustering result contains numerous small, yet coherent clusters. This does not necessarily indicate poor clustering quality, especially when the clusters are semantically meaningful. PUR assigns each cluster to the class that is most frequent in the cluster, measuring the accuracy of this assignment by counting the relative number of correctly assigned elements.

7.5.3 (iii) - Graded Change Detection

Given a word w, we split its DWUG into two subgraphs representing nodes from the two time periods (see Figure 7.1) and quantify the semantic change of w by computing the $\sqrt{JSD}$ between the two time-specific cluster distributions. In contrast, for RU, we adhere to the RuShiftEval procedure and quantify semantic change through the application of the COMPARE metric that directly measures the mean relatedness of annotated word usage pairs as semantic change scores (Schlechtweg et al., 2018). Our evaluation is based on the GCD task and thus we the use Spearman correlation as evaluation metric between predicted ranking and ground truth rankings.

		EN	**DE**	**SV**	**ES**	**RU**			**NO**		**ZH**	**Avg$_w$**
		$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_2 - C_3$	$C_1 - C_3$	$C_1 - C_2$	$C_2 - C_3$	$C_1 - C_2$	$C_i - C_j$
WiC	BERT	.503	.350	.221	.319	.314	.344	.350	.429	.406	.516	.358
	mBERT	.332	.344	.284	.289	.280	.273	.293	.283	.333	.413	.301
	XLM-R	.352	.289	.255	.288	.212	.250	.251	.317	.261	.392	.272
	XL-LEXEME	**.626**	**.628**	**.631**	**.547**	**.549**	**.558**	**.564**	**.484**	**.521**	**.630**	**.568**
	GPT-4.0	.606	-	-	-	-	-	-	-	-	-	-
	Agreement	*.633*	*.666*	*.672*	*.531*	*.531*	*.567*	*.564*	*.761*	*.667*	*.602*	*.593*
WSI	BERT	.136 / .700	.047 / .662	.023 / .596	.189 / .695	- / -	- / -	- / -	.251 / .771	.247 / .758	.279 / .759	.166 / .702
	mBERT	.067 / .644	.054 / .679	.024 / .648	.228 / .700	- / -	- / -	- / -	.241 / .759	.159 / .753	.172 / .713	.146 / .696
	XLM-R	.068 / .737	.024 / .725	.031 / .680	.164 / .755	- / -	- / -	- / -	.179 / .775	.183 / .715	.279 / .806	.133 / .743
	XL-LEXEME	.273 / .834	**.300 / .788**	**.249 / .766**	**.400 / .820**	- / -	- / -	- / -	**.337 / .806**	**.304 / .808**	**.448 / .836**	**.339 / .810**
	GPT-4.0	**.340 / .877**	- / -	- / -	- / -	- / -	- / -	- / -	- / -	- / -	- / -	- / -
GCD	BERT	.425	.116	.148	.284	.487	.452	.469	.571	.521	**.808**	.422
	mBERT	.120	.205	.234	.394	.372	.325	.408	.290	.454	.737	.357
	XLM-R	.219	.069	.143	.464	.284	.301	.375	.395	.345	.557	.324
	XL-LEXEME	.801	**.799**	**.721**	**.655**	**.780**	**.824**	**.851**	**.620**	**.567**	.716	**.754**
	GPT-4.0	**.818**	-	-	-	-	-	-	-	-	-	-

Table 7.5: Evaluation of contextualized models as computational annotators: Spearman correlation for WiC and GCD, Adjusted Random Index and Purity (ARI / PUR) for WSI. Top score for each approach and benchmark is highlighted in **bold**. Avg is a weighted average based on the number of targets in each benchmark test set. For the sake of comparison, we report the Krippendorff's α score for inter-human annotator *agreement* in WiC (*italic*).

Evaluation results – Table 7.5

(i) - Word-in-Context Our evaluation reveals that pre-trained models such as BERT, mBERT, and XLM-R demonstrate a low average correlation with human judgments (.358, .301, .272). In contrast, XL-LEXEME and GPT-4 emerge as powerful solutions for scaling up and aiding human annotations. For EN, they obtain a moderately strong correlation (.626, .606) with human judgments, only marginally lower than the Krippendorf α human agreement (.633). In particular, XL-LEXEME slightly outperforms a considerably larger model like GPT-4 in terms of parameters, at a considerably lower cost. In contrast to previous cross-lingual evaluation (Conneau et al., 2020) and in line with the finding in Table 7.3, mBERT consistently outperforms XLM-R. However, our results highlight the advantageous use of monolingual BERT models over the multilingual ones, for assessing **(i)** - WiC.

We consider the WiC evaluation to be the most valuable as it involves a direct comparison between computational predictions and human judgments.

(ii) - Word Sense Induction Our evaluation indicates that moderate performance in **(i)**-WiC leads to moderately *low* performance in inferring word sense. We obtain low ARI scores across all models and benchmarks, with XL-LEXEME and GPT-4 exhibiting the highest values. Specifically, GPT-4 outperforms XL-LEXEME (with .340 compared to .273) in ARI for EN. However, we highlight that even such low scores represent a moderately *high* result, given an inter-annotator agreement of .633.

XL-LEXEME consistently demonstrates high PUR scores across all benchmarks, while other models yield slightly lower PUR scores, suggesting that some word sense patterns are captured when using contextualized models. Previous studies highlight that contextualized models tend to produce a large number of clusters (Martinc et al., 2020b; Periti et al., 2022), thereby influencing PUR scores. Therefore, it is crucial to interpret PUR in conjunction with ARI.

(iii) - Graded Change Detection As for GCD, we obtain average results for BERT, mBERT, XLM-R, and XL-LEXEME equal to .422, .357, .324, .754, respectively. These results are consistent with those presented in Table 7.3, when compared to form-based approaches (.316 – .751). We observe that employing more word usage pairs, as in Table 7.3, proves beneficial for certain benchmarks in the GCD tasks (e.g., XL-LEXEME+APD for EN and DE). However, we note that these results for **(ii)** - WSI are significantly higher than those obtained by sense-based approaches (.077 – .422). This can likely be attributed the fact that here we are using the same clustering algorithm that was used for obtaining the ground truth clusters, or to the fact that the clustering algorithm is more able to capture nuanced word meaning than AP and APP. In contrast, for RU, following the RuShiftEval procedure does not improve the performance and results between Table 7.3 and 7.5 are somewhat comparable.

7.6 Discussion and considerations

We have performed a first-ever evaluation of models and approaches for modeling LSC under equal settings and conditions, over eight different languages. First, we evaluated different models combined with standard approaches to the popular GCD task. In particular, we consider BERT, mBERT, XLM-R, XL-LEXEME as pre-trained models, APD and PRT as form-based approaches, and AP+JSD and WiDiD as sense-based approaches. We find that the XL-LEXEME consistently outperforms other models across all approaches, and thus should be used as the de facto standard. We also find that form-based approaches significantly outperform sense-based approaches, with APD as the best approach for GCD. Among the sense-based approaches, we find that *evolutionary* clustering is advantageous in contrast to static clustering and should be a focus of future work. We additionally extended the evaluation to include the WiC and WSI tasks, both inherently crucial to solve the complex task of LSC. We compare GPT-4 to the previous models and find that GPT-4 and XL-LEXEME both perform close to human-level while the other models obtain only low-moderate performance. However, due to the considerable costs associated with utilizing GPT-4, extending its evaluation to additional languages is not affordable. In particular, our evaluation reveals that GPT-4 obtains comparable performance to XL-LEXEME. In contrast to the limited accessibility[5] and high associated cost[6] of GPT-4, XL-LEXEME is a considerably smaller, open-source model. Thus, since XL-LEXEME obtains results close to those of GPT-4, even beating it for the WiC task, we argue that the use of GPT-4 is not justified

[5] https://platform.openai.com/docs/guides/rate-limits

[6] https://openai.com/pricing

for modeling the LSC problem and that XL-LEXEME can be used for LSC tasks as a affordable, scalable solution.

All in all, considering the current state of the LSC modeling, we argue that **only obtaining state-of-the-art performance on GCD does not solve the LSC problem**, as there is a clear need to **distinguish the different senses of a word and how these evolve over time**. As stated in Chapter 4, GCD maintains relevance for identifying words that have changed across multiple time periods in need of further *sense-based* modeling. GCD also serves to quantify the change on the level of vocabulary. In conclusion, in this chapter, we provide a first comparable evaluation of contextualized word embeddings for LSC and establish clear settings that should be used for future comparison and evaluation. With this work, we want to raise awareness of the current trend of the community in modeling only the GCD task. Our aim is to shift the focus from merely assessing *how much* to *how*, *when*, and *why*, prompting the development of both *unsupervised* and *supervised* approaches for addressing the full spectrum of LSC.

Limitations. There are limitations we had to consider in the making of our evaluation. Firstly, we could not evaluate GPT-4 across all languages due to both price and API limitations. This means that while the results are comparable with XL-LEXEME for EN, we do not know how GPT-4 will behave for the other languages. Our decision to use GPT-4 over the cheaper GPT-3 is based on recent studies showing conflicting results across different tasks. Notably, Karjus (2023) reported high scores for GPT-4 in the GCD task. However, Periti et al.; Laskar et al.; Kocoń et al. (2024d; 2023; 2023), as well as ourselves in Chapter 3, reported low scores for the WiC task when employing GPT-3. As a result, we opted for GPT-4 to ensure relevance and accuracy in our evaluations.

In our comparison, we evaluate different contextualized models utilizing the popular Transformers library for deep learning maintained by Hugging Face (Wolf et al., 2020). We specifically excluded the evaluation of a BERT model for Latin, opting instead to focus on mBERT, XLM-R, and XL-LEXEME. At the beginning of our evaluation, we were not aware of any experiments using Latin BERT models to address GCD, nor were we aware of an open BERT version for Latin on the Hugging Face platform. As we have only recently become aware of novel BERT models that are exclusively trained and fine-tuned for Latin (Riemenschneider and Frank, 2023; Lendvai and Wick, 2022), we plan to further test and utilize these models in future work.

To make a fair comparison between different contextualized models, we employed the same procedure across all benchmarks and languages. However, different languages have different structures and hence different requirements. It would be equally fair to have different processing of the different benchmarks (e.g., lemmatization for German, Laicher et al., 2021). We opted to reduce the number of open variables to be able to make this first evaluation. Future work could optimize each language and then compare performance.

Lastly, the models compared in this study, despite sharing similar architectures, tokenize text sequences differently based on their reference vocabulary. Consequently, a word may be split into different sub-tokens by one model and represented as a single token by another (Jenkins et al., 2023). Additionally, when contexts exceed the maximum input size, different models may truncate them at various points. Adhering to standard procedures in the field of LSC, we use the average embeddings of sub-words when a word is split into multiple sub-words. However, the impact of different tokenization and truncation methods was not evaluated.

Chapter 8

Analyzing semantic change through lexical replacements

> "*For example, the male/female relationship is automatically learned, and with the induced vector representations,* `King` - *`Man`* + *`Woman`* *results in a vector very close to* `Queen`."
>
> Mikolov et al., In Proc. of NAACL 2013

8.1 Introduction

The major advancement that novel LLMs have brought is the ability to dynamically generate contextualized representations (i.e., embeddings) based on specific usage contexts. When words are used in contexts similar to those encountered during training, LLMs can easily differentiate, in a computational way, between word meanings. Like in the case of *rock* in the sentences *sitting on a rock* and *listening to rock*.

However, when an existing word in our vocabulary gains a new meaning through semantic change, LLMs' ability to differentiate that meaning can be affected. This stems from the fact that semantic change is evidenced through new contexts that were previously unknown for the word. Sometimes, the new meaning is novel to the dictionary, for example, the metaphorical Web-meaning of *surfing*. Other times, the meaning is already in existence and gets the word as a new referent. This is, for example, the case for *happy*. It used to mean exclusively `to be lucky` and then gained the meaning of `happiness`. In an inverse process, the word *gay* lost its meaning of `happiness` and began to refer exclusively to `homosexuality`. One can think of this process of *semantic change* to be a *lexical replacement* of the word *happy* into the context of *gay*, like in the following sentence.

“The heart is sportive, light, and **gay**, life seems a long glad summer’s day”[1]

When using LLMs, the representation of a word w is based on

- (i) the pre-trained knowledge that the model has about w given its position in the context, and
- (ii) the context c in which w is used.

Thus, when this replacement happens, LLMs experience a *tension* between the **existing** sense/s of *happy* (which do not include `happiness`) and the meaning of the **new** context (which does indicate `happiness`). Due to semantic change, LLMs do not know the relationship between the new context c and the replacement word r. As a consequence, the representation of r (i.e., *happy* in the sense `to be lucky`) and the representation of c (i.e., the context of *gay* in the sense of `happiness`) pull in different directions challenging the LLMs’ ability to contextualize (Ethayarajh, 2019).

The tension increases as the gap between the data used for training the model, and the data on which the model is applied grows larger. Indeed, the LLMs we use serve as the lens through which we view the studied texts: if our texts are contemporary with the pre-training, the gap is likely to be minimal. If, however, we intend to study historical or other out-of-domain corpora through LLMs trained on modern text, this gap can be arbitrarily large and have major effects on follow-up studies. Thus, using LLMs for modeling relationships beyond their pre-trained knowledge will likely result in an underestimation of semantic change.

Chapter outline.

This chapter includes materials originally published in the following publication:

> Francesco Periti, Pierluigi Cassotti, Haim Dubossarsky, and Nina Tahmasebi. 2024**b**. Analyzing Semantic Change through Lexical Replacements. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4495–4510, Bangkok, Thailand. Association for Computational Linguistics.

In this chapter, we propose a replacement schema to study the tension experienced by LLMs when words undergo semantic change. Such schema involves replacing a word w in the context c with a replacement r to analyze how the representation of r differs from the original representation of w.

Given a word w, our experiments systematically show that LLMs (i.e., BERT, mBERT, XLM-R) experience a tension between the pre-trained knowledge of w and the new context of a gained meaning. This tension differs across linguistic relations, namely synonymy, antonymy, and hypernymy.

We then use the introduced schema for detecting semantic change. Our experiments show that, when random replacements are used to simulate *synthetic* semantic change, the use of a clustering algorithm (i.e, Affinity Propagation) falls short to differentiate meanings and detect such change. Furthermore, we use

[1] Manchester Times, Wednesday 03 May 1854, found via `https://discovery.nationalarchives.gov.uk`.

the replacement schema to introduce a new *interpretable* model for semantic change detection, while being comparable with state-of-the-art for English.

Finally, we evaluate the use of a predefined set of **lexical replacements** derived from lexicographic resources (i.e., WordNet and Wiktionary) through the LSC task.

In Chapters 5, 6, and 7, we investigated the use of **word embeddings** for LSC. With more computational resources available and the increasing attention on open, generative LLMs such as LLaMa (Touvron et al., 2023a), we decided to compare LLaMa 2 (Touvron et al., 2023b) to BERT for modeling semantic change through automatically generated **lexical substitutes**. Our experiments show that LLaMa 2 significantly outperforms a model like BERT, which is specifically trained to provide lexical substitutes through masked language modeling.

The chapter is organized as follows. Section 8.2 frames our study within the relevant literature of its time. Section 8.3 introduces the replacement schema and the data used in our study. Section 8.4 discusses tension caused by semantic change in LLMs. The implications of this tension for the computational modeling of semantic change are discussed in Section 8.5. In Section 8.6, we present a novel approach to LSC through lexical replacements and compare it with a more approach based on lexical substitutes. Finally, a concluding discussion of our experiments is provided in Section 8.7.

8.2 Related work

For this chapter, relevant work pertains both to the contextualization of modern LLMs and the field of lexical semantic change. Modern *contextualized* LLMs leverage the Transformer architecture to capture the semantics of words (Vaswani et al., 2017). Their success in solving NLP tasks has prompted numerous studies to explore the nature and characteristics of their *contextualization* ability. Ethayarajh (2019); Reif et al. (2019); Cai et al. (2021); Jawahar et al. (2019) shed light on the geometry of the embedding space. Serrano and Smith (2019); Bai et al. (2021); Guan et al. (2020) investigate the interpretability of the attention mechanism. Yenicelik et al. (2020); Garí Soler and Apidianaki (2021); Kalinowski and An (2021); Haber and Poesio (2021) examine the clusterability of word representations. Abdou et al. (2022); Hessel and Schofield (2021); Mickus et al. (2020); Wang et al. (2021a) analyze the impact of word position in the embeddings generation. Reif et al. (2019); Levine et al. (2020); Pedinotti and Lenci (2020) study how word meanings are represented in the embedding space.

Most of the current work involves probing tasks, as proposed by Hewitt and Liang (2019). These tasks consist of training an auxiliary classifier on top of a model, where the contextualized embeddings serve as features to predict syntactic (e.g. part-of-speech) and semantic (e.g. word relations) properties of words (Clark et al., 2019; Lin and Ng, 2022; Wallat et al., 2023; Lin and Ng, 2022; Ravichander et al., 2020). If the auxiliary classifier accurately predicts a linguistic property, the property is assumed to be encoded in the model.

Recent work has focused on a related aspect, namely adapting LLMs to improve their *temporal* contextualization. This challenge has been addressed across various applications such as named entity recognition (Rijhwani and Preotiuc-Pietro, 2020), fake news detection (Hu et al., 2023), text summarization (Cheang et al., 2023), and lexical semantic change (Su et al., 2022; Rosin et al., 2022; Rosin and Radinsky, 2022). Nonetheless, while temporal domain adaptation can improve performance across various tasks, Agarwal and Nenkova (2022) demonstrated that temporal contextualization may not always be a concern.

In this chapter, we complement existing research by using **lexical replacements** as a proxy to analyze how language models contextualize words that have undergone lexical semantic change. Specifically, our work is related to the novel substitute-based approaches to LSC, which interpret word meaning by generating substitutes of words in context (Kudisov and Arefyev, 2022; Arefyev and Zhikov, 2020; Card, 2023). On one hand, word substitutes represent relevant keywords to aid the interpretation of senses. On the other hand, the generation process can only provide substitutes according to training data.

To this end, we propose a novel *interpretable* approach based on a pre-defined set of lexical *replacements* rather than generated *substitutions*.

8.3 Methodology

In our experiments, we leverage a replacement schema to investigate the tension experienced by pre-trained LLMs due to semantic change. This involves analyzing the variations in embedding representations when a target replacement is introduced. For instance, by replacing a target like *cat* with a replacement like *chair* in a specific context like:

$$\text{The } \underset{\text{target}}{\textit{cat}} \leftarrow \underset{\text{replacement}}{\textit{chair}} \text{ was purring loudly .}$$

8.3.1 The replacement schema

We use WordNet to generate different classes of replacements for a specific word (Fellbaum, 1998), which correspond to a varying degree of plausibility (i.e. suitability of a specific replacement) between the target word and its replacement. Thus, we hypothesize that each class is associated with a different impact on contextualization. Each class of replacements also has diachronic relevance, as the synchronic, semantic relation can be considered to have a parallel in semantic change (de Sá et al., 2024; Wegmann et al., 2020). To ensure accurate linguistic replacements, we maintain part of speech (PoS) agreement with the target words; e.g., *nouns* are replaced with *nouns* and so forth. Examples are given in the form (target ← replacement) in Table 8.1.

- **synonyms** (e.g. *sadness* ← *unhappiness*) are used to evaluate the stability in contextualization; that is, we hypothesize similar embeddings between target and replacement words. Indeed, synonyms are considered equally likely alternatives in LM's pre-trained knowledge. On the diachronic level, they emulate the absence of any semantic change of the replacement word;

- **antonyms** (e.g. *hot* ← *cold*) are used to evaluate a light change in contextualization; that is, we hypothesize slightly less similar embeddings between target and replacement words. Indeed, antonyms are sometimes equally plausible alternatives, for example: "I *love/hate* you". Other times they are likely to surprise the model. For example: "I burned my tongue because the coffee was too *hot/cold*". On the diachronic level, they emulate a contronym change. A contronym change occurs when a word's new meaning is the opposite of its original meaning (e.g. *sanction* in English) of the replacement word;
- **hypernyms** (e.g. *animal* ← *bird*) are used similarly to `antonyms`. However, on the diachronic level, they emulate a broadening semantic change of the replacement word;
- **random** words (e.g. *sadness* ← *eld*) are used to evaluate a change in contextualization. If LLMs place high importance on the context, then the replacement should receive a similar representation to the target word. Otherwise, if LLMs heavily rely on its pre-trained knowledge, the replacement will exhibit dissimilarity to the target word despite the identical context, as well as dissimilarity to the typical replacement representations. On the diachronic level, `random` emulates the presence of strong semantic change of the replacement word, that is, the emergence of a homonymic sense.

Lexical Replacement	w^{-4}	w^{-3}	w^{-2}	w^{-1}	w / $w^{(r)}$	w^{+1}	w^{+2}	w^{+3}	w^{+4}
Target Word	moments	of	regret	and	*sadness*	and	guilty	relief	.
Synonym	moments	of	regret	and	*unhappiness*	and	guilty	relief	.
Hypernym	moments	of	regret	and	*feeling*	and	guilty	relief	.
Antonym	moments	of	regret	and	*happiness*	and	guilty	relief	.
Random	moments	of	regret	and	*eld*	and	guilty	relief	.

Table 8.1: Different classes of replacements.

8.3.2 Data

To avoid introducing noise into our experiments resulting from the conflation of senses, we replace words with contextually appropriate replacements based on the intended sense of the word within a specific sentence (e.g, *stone* and *music* for *sitting on a rock* and *listening to rock*, respectively). We therefore leverage the SemCor dataset (Miller et al., 1993), still the largest and most commonly used sense-annotated corpus for English. To select candidate replacements, we consider different PoS tags, namely *verbs*, *nouns*, *adjectives* and *adverbs*, and semantic classes, namely *synonyms*, *hypernyms* and *antonyms*. We randomly sample a set of synsets for each PoS tag occurring in SemCor, and for a specific synset, we extract a subset of contexts (i.e., sentences) where a word is annotated with that synset. We sample a maximum of 10 sentences per synset to prevent oversampling of high-frequency synsets. We control for the position of the replaced target has in the sentence, and the length of the sentence, to confirm that these aspects will not bias our experiments differently across PoS. For each sentence, we generate the *synonym* and *antonym* replacements for all PoS, and *hypernym* replacements only for nouns and verbs because WordNet lacks hypernym information for other

PoS (see Table 8.2).

PoS	**N. target words**	**Avg. N. of sampled senteces per target word**	**N. examples**
noun	360	3.55	1277
verb	433	3.45	1494
adjective	393	3.39	1334
adverb	158	3.46	546

Table 8.2: Data statistics over PoS, sampled from SemCor.

Experimental setup We begin by studying the tension that occurs as a consequence of replacement focusing on the word contextualization in Section 8.4. Next, we the use of replacements as a proxy for semantic change in Section 8.5 and 8.6. In our experiments, we use monolingual BERT[2], mBERT[3], and XLM-R[4]. Our code and data are available at `https://github.com/ChangeIsKey/asc-lr/`.

8.4 Tension caused by semantic change

We analyze the tension experienced by LLMs by comparing the embedding of a target word w in the original sentence c to the embedding of the replacement word r in the same sentence c. To perform this comparison, we rely on the cosine distance between the embeddings of w and r. We refer to this as the *self-embedding distance* (SED).

Concretely, if w and r are split into multiple sub-words by the model, we calculate the average embeddings of the corresponding sub-words. This approach ensures the preservation of the same number of tokens in the original and synthetic sentences and enables accurate distance calculations.

The less plausible the relationship between the context c and the replacement word r for LLMs, the higher the SED, leading them to rely on the pre-trained knowledge of r to contextualize r in context. When there is a large mismatch between the meanings of the replacement word r and the context c, as is the case with the random replacement, then the SED is the highest.

8.4.1 Self-embedding distance

For each pair of original and synthetic sentences, we computed SED across each layer. We then analyzed the average SED for each class of replacement and PoS across the layers of LLMs. It is known that contextualized embeddings experience an anisotropic nature, that is, the embeddings occupy an increasingly narrow cone within the vector space (Ethayarajh, 2019). This means that embeddings, and thus SED scores across layers, are not comparable. To address this issue and thus compare SED both across layers and PoS, we use a layer-specific normalization factor.

[2] *bert-base-uncased*

[3] *bert-base-multilingual-cased*

[4] *xlm-roberta-base*

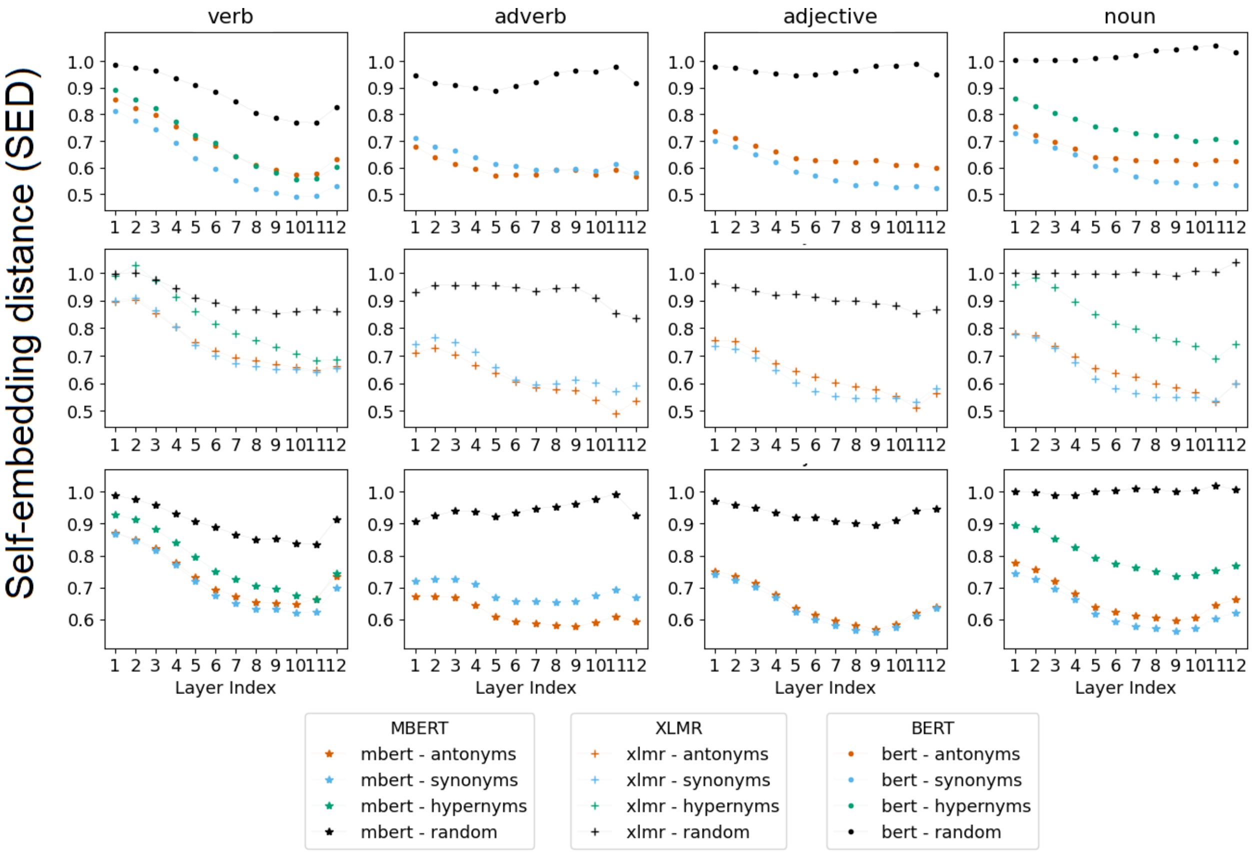


Figure 8.1: Average SED over layers.

Specifically, for normalization, we randomly sampled an additional set of 3864 sentences independent from the sets in Table 8.2. For each sentence, we randomly choose a target word and replace it with a `random` replacement regardless of the PoS agreement. Then, for each layer, we computed the average SED over this set of replacements. We use the resulting SED scores as a normalization factor for each layer that represents an upper-bound approximation. Thus, for each layer, the same normalization factor is used across all PoS and semantic classes of replacement. This way, the normalization cannot influence the discrepancies among different classes for a specific layer but serves to make the scores in different layers somewhat comparable.[5]

Like Ethayarajh (2019), we observe that the contextualization increases across layers as the SED decreases, the context thus has a larger effect in determining the representation of a word in the higher layers. For `adverbs`, `adjectives` and `nouns` the synonym and antonym classes are associated with a SED of around 0.6–0.8 in the first layer. The SED then decreases to between 0.5–0.6. For `adverb` the synonym and antonym class remain similar also in the later layers, while for `adjectives` and `nouns` we find that the synonyms have lower SED than do antonyms. For `nouns`, the hypernym class has consistently higher SED than synonyms and antonyms, despite being a more general concept where the subconcept of the target word

[5] We have tested with different normalization factors – e.g., replacing a word with a special token (“`[REPL]`”) outside the LLMs vocabulary – and found that the conclusions remain.

should be contained (e.g., *fruit* as a hypernym of *banana*). This aligns with the recent findings of Hanna and Mareček (2021), suggesting that BERT's understanding of noun hypernyms is limited.

The SED score for `random` is fairly stable across all layers, meaning that when a word gains a completely novel sense, LLMs fall short in contextualizing beyond the pre-trained knowledge it has of the word. That is, the representation of the random word does not mimic the representation of the target word that it replaces. The context thus has little or no effect in determining the representation of the replacement word.

For `verbs`, we note a higher SED for antonyms and synonyms in comparison to other PoS, comparable to the `noun` hypernyms, starting around 0.9. However, they all drop to 0.6–0.7 by the last layers. Additionally, there is a narrower gap between the SED for the random class and those for antonyms, synonyms, and hypernyms. These observations suggest that, in the earlier layers, the contextualization of verbs is less pronounced for `verbs` and that the model relies more on pre-trained knowledge.

All in all, our results suggest that models exhibit varying tension for different PoS, and for different linguistic relationships between the target and the replacement word. Conversely, we interpret these findings in the following way: there is a low degree of contextualization, and thus a high degree of tension, when there is no relationship between the word and its replacement.

8.5 Semantic change

We argue that our findings in Section 8.4 regarding the tension between a word and its context have important implications when pre-trained LLMs are used for modeling semantic change as we will show in this section.

8.5.1 LCS through synthetic dataset

Form-based approaches can still detect this semantic change to a certain degree (as an estimate of model confusion), despite using contextualized word embeddings that do not correctly capture a word's meaning in a novel context. However, *sense-based* approaches fall short in accurately detecting the same change. This is because *sense-based* approaches require modeling meanings outside the model's pre-trained knowledge before detecting the change. Since these meanings cannot be adequately modeled when semantic change has occurred, the performance of *sense-based* approaches is reduced compared to that of *form-based* approaches.

We further tested these implications in the LSC task by comparing PRT (based on *averaging* contextualized embeddings) and JSD (based on *clustering* contextualized embeddings) on an artificial diachronic corpus spanning two time periods (see details in Appendix B). Essentially, we introduced random replacements in C_2 with varying probabilities to emulate different degrees of change for a set of 46 target words. Subsequently, we compared the Spearman Correlation between the scores obtained with PRT and JSD with the artificially graded score of emulated semantic change. Results using BERT are presented in Figure 8.2 (see Appendix C for additional results). Our hypothesis is that while PRT can predict changes to a fairly high degree, JSD falls short because it can only model the meanings that BERT is already aware of.

As shown in the figure, using PRT, we can model artificial semantic changes already from layer 3. This is

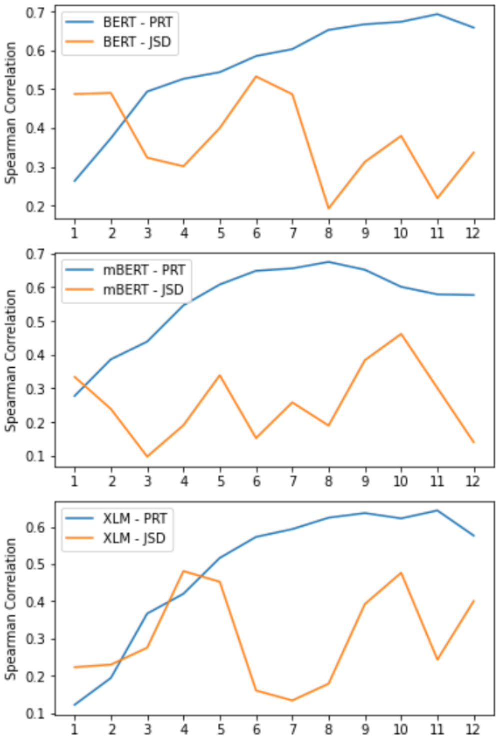


Figure 8.2: Spearman Correlation over layers for artificial semantic change.

not the case for JSD, where we observe statistically significant correlations for only a few layers. However, the significance of performance for JSD is an artifact of BERT embeddings and does not authentically represent the simulated change. We verify this by examining the modeled clusters. While, in general, the number of clusters of AP is large (Periti et al., 2022; Martinc et al., 2020b), representing *sense nodules*[6] rather than word meanings (Kutuzov et al., 2022b), we find that the injected confusion in the model due to the `random` replacements results in a very low number of clusters (typically 2, maximum of 4). We report similar results in Figure 8.3 for other languages (i.e. German, Swedish, Spanish)

8.6 A novel approach to LSC through replacements

We propose a novel supervised approach to Graded Change Detection building upon the replacement schema. Our approach leverages a curated set of word replacements from WordNet and Wiktionary.

We denote $T = \{w_1, w_2, ..., w_N\}$ as the set of target words. For each target word, we extract a set of possible replacements $\rho(w_i) = \{r_1, r_2, ..., r_M\}$, resulting in $N \cdot M$ replacement pairs. The set of replacements is obtained by considering the lemmas of synonyms and hypernyms associated with the target word w_i in WordNet and words extracted from the Wiktionary page corresponding to the target word. For each target word w_i, we sample up to 200 sentences from each period that remain stable regardless of the replacement

[6] "Lumps of meaning with greater stability under contextual changes" (Cruse, 2000)

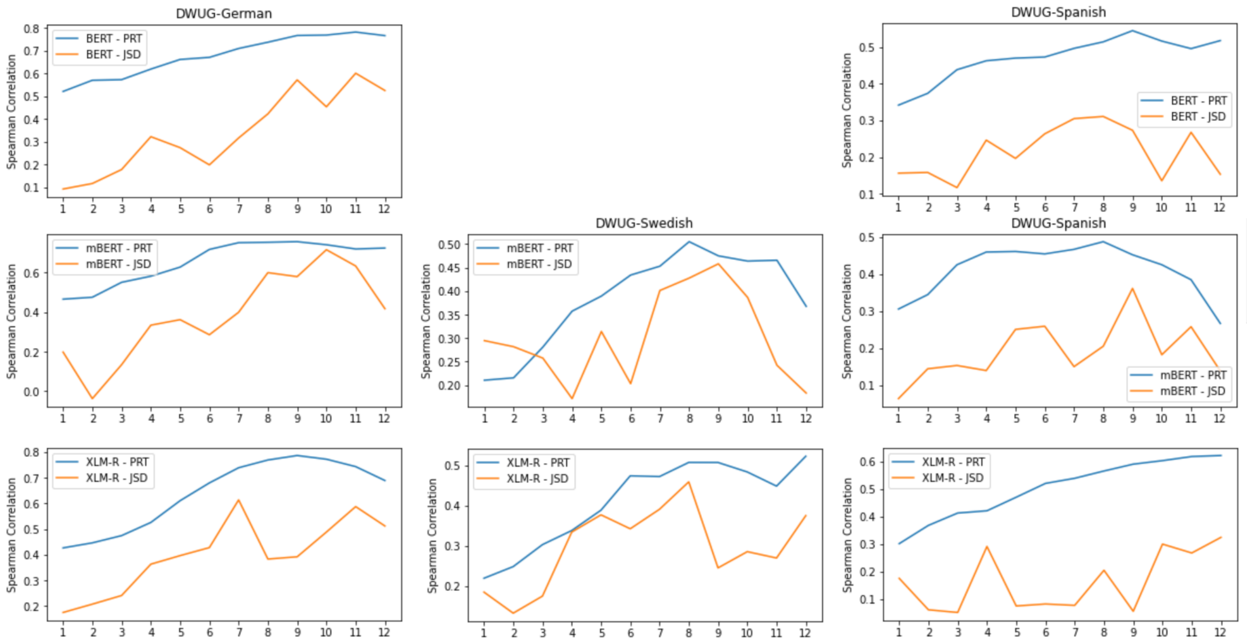


Figure 8.3: PRT and JSD performance on the artificial LSC dataset

word r_j. For each replacement pair (w_i, r_j), we denote the set of sentences for a time period $t \in \{1, 2\}$ as $S^t(w_i, r_j)$.

For each sentence $s \in S^t(w_i, r_j)$ we measure the self-embedding distance $sed(s)$of the target and replacement word. The average self-embedding distance of a target-replacement pair is defined as

$$awd^t(w_i, r_j) = \frac{1}{|S^t(w_i, r_j)|} \sum_{s \in S^t(w_i, r_j)} sed(s)$$

The absolute difference in awd over time is denoted $\mathrm{TD}(w_i, r_j)$. Finally, we rank the replacements $\rho(w_i)$ according to their degree of time difference:

$$R(\rho(w_i)) = \{r_1, r_2, ..., r_M \,|\, \mathrm{TD}(w_i, r_{i+1},) \leq \mathrm{TD}(w_i, r_i)\}$$

and we compute a semantic change score lsc_w as the average TD considering the top k replacements:

$$lsc_w = \frac{1}{k} \sum_{r \in R(\rho(w_i))_k} \mathrm{TD}(w_i, r)$$

We evaluate our approach on the SemEval-2020 Task 1, Subtask 2 dataset for English. We compute the Spearman Correlation between the graded score reported in the gold truth and the lsc scores. Figure 8.4 reports the correlation computed for different values of k. The highest correlation of 0.741 is achieved when considering the first 22 replacements, while the lowest correlation of 0.600 is obtained using only the first replacement (see Table 8.4). Interestingly, the minimum correlation obtained using the replacements

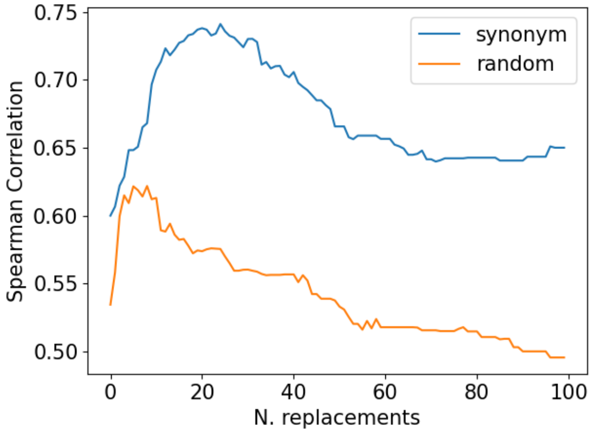


Figure 8.4: Top-k replacements vs Spearman Correlation.

is competitive with SOTA results. Moreover, on average, the correlation is higher than the SOTA model's performance. The replacements are reported in Table 8.3.

By replacing the target words with different semantically related words, we generate contextual variations that enable the detection of semantic shifts. In the case of words like *record* (attainment, track record ⟶ evidence, document) and *land* (real estate, real property ⟶ realm, country) that have undergone semantic change through narrowing and generalisation, respectively, linguistically aware replacements can provide valuable insights. The replacement process generates a list of replacements that can be used as labels for the types of semantic change observed. By associating each replacement with a specific semantic category or change type, it becomes possible to analyze and quantify the semantic shifts experienced by words over time. The method can also be combined with a priori clustering to get changes specific to a sense.

Random replacements Here, we focus on the results using randomly selected words with the same PoS as the target word, i.e. `random` replacement as introduced in Section 8.3. This approach generates a list of replacement words contextually unrelated to the target word. Some interesting patterns emerge when these results are compared with those obtained using synonym replacement. In the case of semantic change detection, the use of synonyms can provide more contextually relevant replacements, as they share semantic relationships with the target word. However, using random replacements can still yield reasonable results, as evidenced by an average correlation of 0.542. These results is in line with the finding of Section 8.5.

In this approach, although random replacements tend to perform worse than synonym replacements, they have one distinct advantage: they do not rely on external lexical resources and are thus suitable for unsupervised scenarios. While synonym replacements can improve contextualization and semantic relevance, they are not always readily available or reliable for languages with limited linguistic resources. In such cases, random replacements can still provide reasonable results and serve as a practical, resource-efficient approach for tasks where synonym information is scarce or unavailable.

Word	Time span	(Ranked) Farthest replacements	lsc_w (k=1)
attack	T1	**physical**, degeneration, blast, crime, disease, death, condition, plane, affliction, birthday attack	-0.036
	T2	**approach**, force, onslaught, assault, exploit, challenge, commencement, aim, worth, signal	0.059
bit	T1	**nominative case**, accusative case, cryptography, information theory, bdsm, time,point, binary digit, sociologic, sublative	-0.018
	T2	**saddlery**, chard, illative case, iron, bevelled, tack, small, gun, cut, elative case	0.067
circle	T1	**wicca**, circumlocution, encircle, astronomy, tavern, semicircle, around, logic, go,wand	0.002
	T2	**pitch**, place, graduated, figure, disk, territorial, enforce, worship, line, bagginess	0.064
edge	T1	**brink**, cricket, instrument, margin, polytope, side, edge computing, verge, demarcation line, demarcation	-0.015
	T2	**data**, production, climax, division, superiority, organization, sharpness, graph, win, geometry	0.047
graft	T1	**lesion**, bribery, felony, politics, bribe, corruption, autoplasty, surgery, nautical, illicit	-0.047
	T2	**branch**, stock, tree, fruit, shoot, join, cut, graft the forked tree, stem, portion	0.103
head	T1	**headland**, head word, capitulum, syntactic, pedagogue, fluid dynamics, hip hop, headway, pedagog, word	0.004
	T2	**leader**, organs, implement, top, tail, foreland, chief, bolt, axe, forefront	0.084
land	T1	**real estate**, real property, surface, property,build, physical object, Edwin Herbert Land, electronics, landing, first person	-0.032
	T2	**realm**, country, kingdom, province, domain, people, homeland, territory, nation, region	0.076
lass	T1	**sweetheart**, girl, missy, woman, yorkshire, lassem, lasst, lassie, loss, miss	0.014
	T2	**fille**, dative case, jeune fille, loose, lasses, unattached, young lady, young woman, north east england, past participle	0.099
plane	T1	**airplane**, aeroplane, pt boat, heavier-than-air craft, glide , boat, lycaenidae, lift, bow, hand tool	-0.197
	T2	**geometry**, point, shape, surface, flat, degree, form, range, anatomy, smooth	0.205
player	T1	**media player**, idler, soul, thespian, person, individual, trifler, performer, somebody, histrion	-0.065
	T2	**contestant**, performing artist, actor, musician, musical instrument, music, gamer, theater, player piano, play the field	0.042
prop	T1	**props**, airscrew, astronautics, actor, airplane propeller, seashell, stagecraft, stage, property, art	-0.042
	T2	**around**, rugby, imperative mood, about, singular, scrum, ignition, roughly, ballot, manually	0.088
rag	T1	**ragtime**, nominative case, accusative case, rag week, terminative case, inflectional, sublative, piece of material, tag, sanitary napkin	-0.049
	T2	**clothes**, exhaustion, university, society, silk, ragged, journalism, haze, ranking, torment	0.071
record	T1	**attainment**, track record, achievement, accomplishment, struct, number, intransitive, record book, criminal record, disc	-0.036
	T2	**evidence**, document, information, audio, recollection, storage medium, memory, electronic, sound recording, data	0.089
stab	T1	**thread**, staccato, feeling, nominative case, sheet, chord, bacterial, culture, twinge, sensation	-0.046
	T2	**wound**, tool, knife thrust, weapon, plaster, criticism, wire, pierce, thrust, try	0.029
thump	T1	**clunk**, throb, clump, thud, pound, thumping, rhythmic, sound, blow, hit	-0.036
	T2	**muffled**, hit, blow, sound, rhythmic, thumping, pound, thud, clump, throb	0.033
tip	T1	**gratuity**, first person, forty, bloke, singular, overturn, stringed instrument, unbalanced, taxi driver, sated	-0.031
	T2	**brush**, tap, strike, gift, tram, flex, tumble, heap, full, hint	0.070

Table 8.3: Words annotated as changed in SemEval 2020 Task 1: Binary Subtask and retrieved farthest replacements for each time span.

	Model	**Spearman Correlation**
	Rosin and Radinsky	0.629
	Kutuzov and Giulianelli	0.605
	Laicher et al.	0.571
	Periti et al.	0.512
	Cassotti et al. (XL-LEXEME)	0.757
Synonym Replacement	Replacement Min. Corr.	0.600
	Replacement Max. Corr.	0.741
	Replacement Avg. Corr.	0.674
Random Replacement	Replacement Min. Corr.	0.495
	Replacement Max. Corr.	0.622
	Replacement Avg. Corr.	0.542

Table 8.4: Spearman Correlation on SemEval-2020 Task 1 (Eng).

In Section 8.4.1, when using SemCor, we effectively account for the nuances of different word senses, thereby improving the contextualization and semantic relevance of synonym replacements. This approach is more targeted as synonyms are selected based on their association with a particular sense, leading to higher quality contextualization in the context of that sense. As a result, synonym replacements are more finely tuned to the specific meaning of the target word, reducing noise and improving correlation with semantic change labels.

8.6.1 Addressing LSC through substitutions

Finally, we assess the use of lexical substitutes generated by LLMs for LSC. By asking LLMs' to generate substitutions, we probe them for their information about the target word given the context. Similar to Card (2023); Arefyev and Zhikov (2020), we use monolingual BERT. We additionally compared the use of a larger, generative model such as LLaMa 2 7B (Touvron et al., 2023b)[7].

For BERT, we use the masking strategy, meaning that we mask a target word with the special token and generate possible substitutes. For LLaMa 2, we fine-tune the model to enable it to predict the target word. Specifically, we fine-tune LLaMa 2 by inputting the original sentence, adding two asterisks at the beginning and end of the target word. Following the sentence we provide the list of substitutes found in ALaSCA (Lacerra et al., 2021), the largest existing dataset for lexical substitution:

During the siege, George Robertson had appointed Shuja ul-Mulk, who was a **bright** boy only 12 years old and the youngest surviving son of Aman ul-Mulk, as the ruler of chitral. **|*answer*|** *intelligent* **|*s*|** *clever* **|*s*|** *smart* **|*end*|**

where **|answer|**, **|s|**, and **|end|** are added as special tokens in the model. For efficiency reasons, we train the model using the QLoRA paradigm (Dettmers et al., 2023). We fine-tuned for one epoch using a learning rate of 2e-4, and set the LoRA configuration with a rank of 8 and an alpha of 16.

The data used for the evaluations is the same in Section 8.6. In Table 8.5 we report an example of the generated substitutions.

[7] *Llama-2-7b-hf*

	T1	**T2**
	remember that it be only such line as be nearer the ground **plane** than the eye that be draw under the horizon line	as his **plane** cross north carolina and head south over the atlantic it pick up a small convoy of escort military craft that try to make radio contact but fail
BERT	there, be, where, here, and	planes, over, out, boats, aircraft
LLaMa 2	level,surface,flat plane,horizontal plane	aircraft,airplane,jet,plane model,propeller-driven vehicle

Table 8.5: Generated substitutions for usages of **plane** extracted by SemEval 2020 Task 1 English.

Model	**Spearman Correlation**
Arefyev and Zhikov, 2020	0.299
Card, 2023	0.547
LLaMa 2 7B	**0.731**
BERT	*0.450*

Table 8.6: Spearman Corr. on SemEval-2020 Task 1 (EN)

To calculate the degree of semantic change, we consider all uses of a word in time periods t_1 and t_2. We consider the substitutes generated for each usage and calculate the distance between all possible pairs of uses between t_1 and t_2. To calculate the distance, we use the Jaccard Distance between the sets of generated substitutes. Lastly, the Jaccard distances are averaged, and we use the average as a score for LSC. In Table 8.6 we show the result on the SemEval 2020 Task 1 - Subtask 2 (other comparable results in Table 8.4). Our results for BERT are somewhat comparable with SOTA results, while being lower to those obtained through lexical replacements, likely because the replacements are of higher quality when found using WordNet, while the substitutions are generated by the model with its limited knowledge of the context. In contrast, our results for LLaMa 2 are even higher than the results obtained with lexical replacements achieving comparable performance to the one obtained with the recent XL-LEXEME model. We attributed this higher performance to the fact that both LLaMa and XL-LEXEME have been fine-tuned on generating lexical substitutes and WiC task, respectively which, rather than using all of the model's pre-trained knowledge, forces the model to focus on the semantic aspect specifically.

8.7 Discussion and considerations

In this chapter, we study semantic change using lexical replacement. From the point of view of the replaced word, a semantic change takes place as the word gains contexts which it has not encountered previously. When the replacement is closely related to the target word, for example by synonymy, the novelty of the context for the replacement word should be low. However, novelty will increase as the relation between the target and replacement becomes more distant. We are assuming that the replacements based on synchronic relations will offer insights into semantic change diachronically.

To test this hypothesis, we used self-embedding distance (SED) when the context stays the same, using all layers of BERT, mBERT, and XLM-R across four PoS. Not surprising, we found that the self-embedding distance is smallest for synonym replacements and highest for the random replacements. And like Ethayarajh (2019), we found that more contextualization happens across the last layers. For different LLMs, we also find slightly different behaviors. However, consistently, adverbs and adjectives have lower SED scores than verbs and nouns. We show that hypernymy is a more distant relation for LLMs than antonymy and synonymy

We then employ replacements for measuring the degree of semantic change. For this, we generate synonym replacements using WordNet, for each word in the English portion of the SemEval-2020 Task 1 benchmark. We assume that if a word has not experienced semantic change, the SED between the replacements and the target word are similar across time. If however, a word has experienced semantic change resulting in context changes, SED scores will be different over time as the replacements will be more distantly related to the contexts. This method offers a novel *interpretable* semantic change detection. Finally, we ask the LLMs themselves to generate substitutions for a target word in the English SemEval data. This experiment shows the LLMs knowledge of the target words and the semantic change they have experienced. All in all, the lexical replacement schema offers a good way to approach semantic change detection, but also to learn more about our LLMs and their ability to handle semantic change.

Limitations. A potential limitation of our study lies in the use of the replacement schema in conjunction with lexical replacements generated from WordNet: inherent limitations of WordNet, such as potential gaps, inaccuracies, or ambiguities in the semantic relationships may influence our analysis. WordNet also limits the data sources from which we can draw sentences, since we need a corpus with sense annotations corresponding to a lexicon.

Furthermore, in our first experiment, the lexical replacement process involves replacing a *word* occurrence in the original sentence with a related *lemma* extracted from WordNet. As a result, providing the model with synthetic sentences containing the lemma instead of the inflected word may influence the generation of word embeddings and the contextualization of every word in the sentences. However, we assume that this limitation equally affects every class we consider and all models. For example, while the lemma of a verb may reduce the third singular verb form, the plural forms of adjectives and nouns can also be simplified to singular lemma forms. Additionally, to mitigate these issues and ensure that all PoS are equally affected by the replacement procedure, we replaced both the target and replacement words with lemmas in the original and synthetic sentences, respectively. We did not analyze semantic change in Section 5 with respect to different PoS because there are no available LSC benchmarks with a substantial number of targets for different PoS, nor any sense-tagged benchmarks except for a small subset for German.

Finding the correct form of a replacement requires advanced morphological analysis and carries the risk of leading to errors. For now, we therefore opted to circumvent this by replacing targets and lemmas alike. Furthermore, we would like to highlight a relevant study by Laicher et al. (2021) that delves into the influence of various linguistic variables on the use of BERT embeddings for the LSC task. This research demonstrates that by reducing the influence of orthography through lemma usage, significant enhancements in BERT's performance were observed for German and Swedish, while maintaining comparable results for English. This underscores the potential benefits of lemma-based contextualization and that linguistic features like orthography can sometimes be minimized without substantial loss of performance.

Unlike our initial experiments using SemCor sentences, the word occurrences considered in the LSC experiments are not associated with manually sense-annotated information. For this reason, we rely on a *lexical replacement* process at a different level of granularity, which involves replacing *all* occurrences of a word with a related *lemma* extracted from WordNet (rather than replacing a specific word occurrence).

We used LLaMa 2 only for our last experiment. This stems from the difficulty to generate contextualized representation of a single word in context in LLMs. We also do not exhaustively test LLMs as this lies outside the scope of the paper, while requiring a lot more resources. Instead, we use one open LLM to test the knowledge of a LLM when trained on significantly more data compared to BERT-like models.

Finally, in the introduction, we use the example of *gay* and *happy* to illustrate that word *happy* is replaced in contexts of *gay* for the meaning of `happiness`. We are however aware that happy gained the meaning of `happiness` several hundred years before *gay* lost its sense of `happiness`, and only use the example for illustrative purposes.

Chapter 9

Automatically generated definitions and their utility for modeling word meaning

> "*Defeated by those practices of consolation, José Arcadio Buendía then decided to build the memory machine that he had desired once in order to remember the marvelous inventions of the gypsies. The artifact was conceived as a dictionary, so that in a very few hours there would pass before his eyes the notions most necessary for life.*"
>
> Gabriel G. Marquez, *One Hundred Years of Solitude*

9.1 Introduction

Modeling *lexical semantics* using unstructured text has a longstanding history in NLP due to its crucial role in both Natural Language Understanding and Natural Language Generation (Karanikolas et al., 2024; Pustejovsky and Boguraev, 1993). Over the past decades, there have been many relevant technological developments: from count-based (Naseem et al., 2021) to static (Mikolov et al., 2013a) and contextualized (Peters et al., 2018) language models, and most recently, generative models (Hadi et al., 2023). Each of these advancements has contributed significantly to the goal of *modeling the meaning of words*.

Modern language models are based on the Transformer (Vaswani et al., 2017) architecture. Given a word, these models generate semantic representations for each occurrence of the word based on its surrounding context (Apidianaki, 2023). Ideally, these representations should be similar for semantically related word usages and different for semantically distinct ones. Typically, *contextualized* vectors (i.e., embeddings, Pilehvar and Camacho-Collados, 2021) or lexical substitutes (i.e., bag-of-words, Arefyev and Zhikov, 2020) are employed to represent word usages. However, recent advancements in text generation are shifting the attention towards representing word usages through generated *sense definitions* (Giulianelli et al., 2023).

Automatically generated sense definitions provide a dual advantage. Firstly, they distill the information stored in a sentence by abstracting away from the context. Their use potentially condenses various word usage representations pertaining to the same underlying meaning. Secondly, generated definitions provide a means to directly interpret word meaning from unstructured text, thereby enabling language models to serve as surrogate for dictionaries when encountering unfamiliar words (Malkin et al., 2021), or known words in unfamiliar settings (Weiland et al., 2023).

Chapter outline.

This chapter includes materials originally published in the following publications:

> Francesco Periti, David Alfter, and Nina Tahmasebi. 2024**a**. Automatically Generated Definitions and their utility for Modeling Word Meaning. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (to appear), Miami, Florida. Association for Computational Linguistics.

In this chapter, we automatically generate definitions for words *in-context* by relying on two fine-tuned variants of the Llama chat models (Touvron et al., 2023a) refined through instruction tuning (Zhang et al., 2024) on lexicographic resources. We call the models `LlamaDictionary` and assess their performance in Definition Generation, achieving new state-of-the-art results on multiple datasets.

We further extend our evaluation by using `LlamaDictionary` and the `Flan-T5-Definition` model fine-tuned by Giulianelli et al. (2023) for large-scale modeling of word meaning. Specifically, we employ the generated sense definitions as intermediate sense representations. These representations are encoded using a pretrained sequence embedding model rather than using standard token embeddings. We assess the use of `LlamaDictionary` and `Flan-T5-Definition` with thirteen `SBERT` models and evaluate our approach on three popular Natural Language Processing tasks, namely Word-in-Context, Word Sense Induction, and Lexical Semantic Change, achieving new state-of-the-art results on all three tasks.

This chapter is organized as follows. In Section 9.2, we provide background information on word representations commonly used in modeling word meaning and an overview of the current state-of-the-art in generating word sense definitions. In Section 9.3, we introduce our `LlamaDictionary` models for automatically generating definitions. In Section 9.4, we present the setup of our evaluation, which encompasses four distinct NLP tasks: Definition Generation, Word-in-Context, Word Sense Induction, and Lexical Semantic Change. In Section 9.5, we discuss the results obtained from each of these evaluation tasks. Finally, in Section 9.6, we discuss the utility of automatically generated definitions for modeling word meaning, as well as the limitations of our work.

9.2 Background and related work

Word usage representations. With the advent of Transformers, we have witnessed the emergence of large language models capable of contextualizing words within diverse contexts. Unlike static models (Pennington et al., 2014), we now rely on a multitude of contextualized embeddings per word. On one hand, this capability represents an invaluable tool for modeling lexical semantics (Petersen and Potts, 2023), as distances between embeddings have proven to be excellent discriminators of word meaning. On the other hand, it poses interpretability challenges, as embeddings tend to represent contextual variance rather than lexicographic senses (Kutuzov et al., 2022b). Further challenges arise from the broad and heterogeneous distribution of semantic structure across embedding dimensions (Senel et al., 2018).

Lexical substitutes are often employed as alternative representations to raw embeddings (Alagic et al., 2018). These representations consist of sets of automatically generated replacements for specific occurrences of words in-context. Unlike embeddings, lexical substitutes can be directly inspected to infer word meaning. However, the interpretation process requires more time and effort compared to the conventional practice of consulting a dictionary for satisfying meaning definitions. Additionally, interpreting the meaning of a word remains challenging, as lexical substitutes can include stopwords and partial word pieces (Card, 2023), equally plausible alternatives with different meanings (Chiang and Lee, 2023), and even contradictory replacements (Justeson and Katz, 1991).

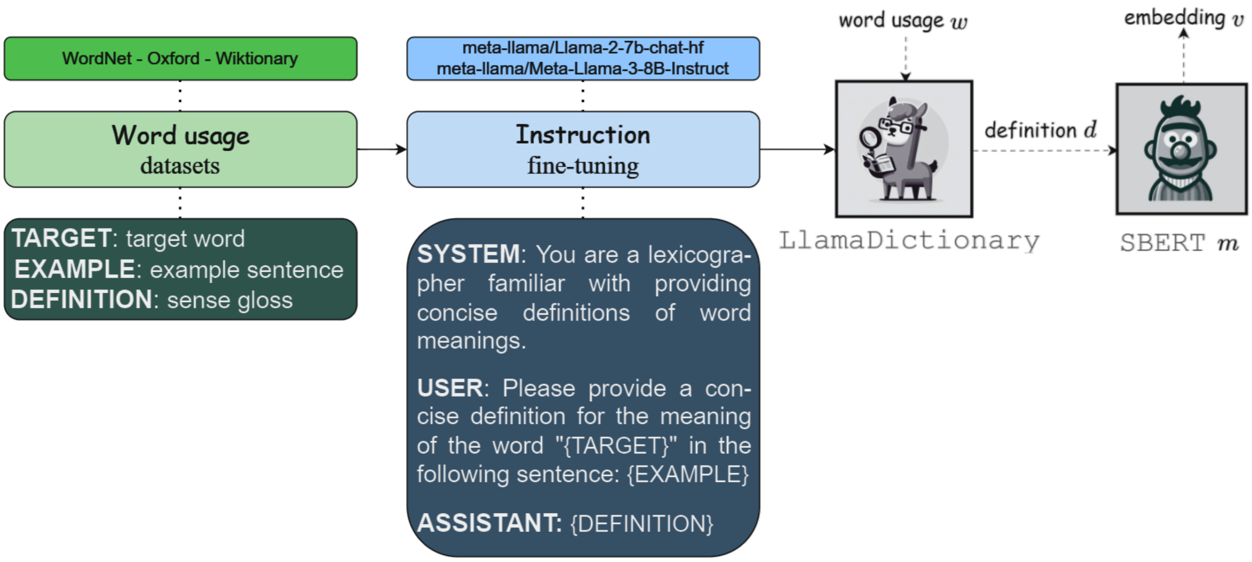
With the recent advancements in text generation, *automatically generated sense definitions* become a viable approach for word usage representation, as these definitions offer descriptive interpretations of words *in-context*, providing a valuable tool with a level of interpretability comparable to manually curated vocabularies (Gardner et al., 2022).

Generating word sense definitions. Generating word sense definitions has initially gained attention to enhance the interpretability of static embeddings (Mickus et al., 2022; Gadetsky et al., 2018). Originally, the task involved generating a natural language definition given a single embedding of a target word (Noraset et al., 2017). However, since words can carry multiple meanings, advancements in contextualized modeling have shifted the focus to the generation of appropriate sense definitions for words in context (Zhang et al., 2022; Huang et al., 2021; Mickus et al., 2019; Ishiwatari et al., 2019).

Generated definitions are useful in a multitude of applications such as the generation of lexicographic resources for low-resource languages (Bear and Cook, 2021), explaining register- or domain-specific vocabulary (Ni and Wang, 2017; August et al., 2022), or language learning scenarios (Zhang et al., 2023; Kong et al., 2022; Yuan et al., 2022).

While early works use sequence-to-sequence models for definition modeling (Ni and Wang, 2017; Gadetsky et al., 2018; Mickus et al., 2019), later works utilize pretrained language models such as BART (Bevilacqua et al., 2020; Segonne and Mickus, 2023; Lewis et al., 2020) and T5 (Huang et al., 2021; Tseng et al., 2023; Raffel et al., 2020).

More recently, Giulianelli et al. (2023) has proposed using generated definitions as interpretable word us-



TARGET: target word
EXAMPLE: example sentence
DEFINITION: sense gloss

SYSTEM: You are a lexicographer familiar with providing concise definitions of word meanings.

USER: Please provide a concise definition for the meaning of the word "{TARGET}" in the following sentence: {EXAMPLE}

ASSISTANT: {DEFINITION}

Figure 9.1: `LlamaDictionary` is a Llama chat model fine-tuned with lexicographic resources to generate a sense definition from an input word usage.

age representation for the analysis of lexical semantic change and fine-tuned a new model called `Flan-T5-Definition` based on Flan-T5 (Chung et al., 2024). Inspired by this work, we follow the idea that definitions can be used as interpretable representations and also position our work with a focus on modeling word meaning and meaning change. Inspired by Bevilacqua et al. (2020), we encode definitions as sentence embeddings. However, we model the meaning of words *in-context* with a single sense definition rather than a set.

9.3 Automatic definition generation

In this chapter, we fine-tuned two popular open-source generative models through instruction tuning, namely Llama2chat[1] and Llama3instruct[2]. We specifically chose to fine-tune chat models because they were already optimized to generate responses adhering to specific instruction prompts. We call the models resulting from fine-tuning `LlamaDictionary`. In the following, we refer to `Llama2Dictionary` and `Llama3Dictionary` for the fine-tuned versions of Llama2chat and Llama3instruct, respectively.

Using `Llama2Dictionary` and `Llama3Dictionary`, we complement the existing `Flan-T5-Definition` 3B model by Giulianelli et al. (2023) with two larger Llama 7B and 8B, chat-based versions.

[1] *Llama-2-7b-chat-hf*
[2] *Meta-Llama-3-8B-Instruct*

9.3.1 Data

We fine-tune Llama2chat and Llama3instruct on the same English data used by Giulianelli et al. (2023). The data consists of *word usages* $\langle w, e, d \rangle$, where w represents a target word, e denotes an example context where w occurs, and d is a human-curated definition for the lexicographic sense of the word w in the example e. The considered word usages span three benchmarks previously extracted from the **Oxford** English Dictionary (Gadetsky et al., 2018), **WordNet** (Ishiwatari et al., 2019), and **Wiktionary** (Mickus et al., 2022), respectively. However, while Giulianelli et al. (2023) use all the Train-Dev-Test partitions during fine-tuning, we use only Train and Dev and reserve Test for evaluation purposes. Table 9.1 reports the main statistics of these benchmarks.

		Oxford	**WordNet**	**Wiktionary**	**Tot.**
Train	# words	33,128	7,935	18,030	45,070
	# definitions	97,802	13,854	31,142	142,798
	# def. per word	2.95	1.75	1.73	3.17
Dev	# words	8,863	998	2,561	11,666
	# definitions	12,222	1,748	4,525	18,495
	# def. per word	1.38	1.75	1.77	1.59
Test	# words	8,848	1,001	2,361	11,718
	# definitions	12,228	1,774	4,436	18,438
	# def. per word	1.38	1.77	1.69	1.57

Table 9.1: Train-Dev-Test partitions of the considered benchmarks. For each partition, we report the number of unique words, the number of unique definitions, and the average number of definitions per target word.

9.3.2 Fine-tuning

Llama2chat and Llama3instruct with 7 and 8 billion parameters, respectively, are large, decoder-only architectures trained on extensive amounts of data, followed by supervised fine-tuning through instruction tuning (Zhang et al., 2024) and iterative refinement using reinforcement learning from human feedback (Kaufmann et al., 2024). We further fine-tuned these models through instruction tuning for sense definition generations.

Given the high costs associated with fine-tuning large language models, we employed a parameter-efficient fine-tuning (Han et al., 2024) that enables efficient adaptation by only fine-tuning a small number of additional model parameters instead of the entire model. This approach significantly reduces computational and storage costs. Specifically, we fine-tuned using Low-rank Adaptation (LoRA, Hu et al., 2021). [3] Experimented hyper-parameters are reported in Table D.1 and D.2.

For fine-tuning, we used cross-entropy loss calculated on all tokens over 4 epochs, with a batch size of 32, a maximum sequence length of 512, and *packing* to train efficiently on multiple samples simultaneously (Kosec et al., 2021).

[3] We provide all our data, code, and results at https://github.com/FrancescoPeriti/LlamaDictionary.

In line with Huerta-Enochian (2024), who demonstrated that prompt loss can be safely ignored for many datasets, we observed lower preliminary results in the evaluation tasks for models chosen based on validation performance. Therefore, we selected the final model based on the checkpoint at the last training epoch.

9.3.3 Instruction-tuning

We fine-tuned Llama2chat and Llama3instruct using the prompt shown in Figure 9.1. For each word usage $\langle w, e, d \rangle$, we substituted `TARGET` with the actual target w, and `EXAMPLE` and `DEFINITION` with the example e and the definition d, respectively.

For our prompt, we drew inspiration from prompts used in previous work, specifically, we employed a prompt similar to those used by Giulianelli et al. (2023). In line with Li et al. (2023), we incorporated an emotional stimulus (in Figure 9.1, `Please`) to enhance the performance. Additionally, similarly to Kocoń et al. (2023); Laskar et al. (2023); Periti et al. (2024d), we structured our prompt in a format that facilitates parsing and comprehension.

9.4 Evaluation setup

Our evaluation is structured into two parts. First, we assess the quality of definitions generated by `Llama Dictionary` and `Flan-T5-Definition` through the Definition Generation (DG) task. For this evaluation, we directly utilize the generated sense definitions.

Next, we explore their utility in three popular Natural Language Processing tasks, namely Word-in-Context (WiC), Lexical Semantic Change (LSC), and Word Sense Induction (WSI). Specifically, instead of using standard token embeddings, we view sense definitions as intermediate sense representations and encode these as embeddings through a pretrained sequence embedding model. Formally, this means that: given an occurrence of a word w, we employ a generative model g (i.e., `LlamaDictionary` or `Flan-T5-Definition`) to generate a definition d, which we subsequently encode as a vector v using a sentence embedding model m, i.e., $v = m(d) = m(g(w))$.

Following Giulianelli et al. (2023), we used the *all-distilroberta-v1* sentence SBERT model (Reimers and Gurevych, 2019) to encode definitions as contextualized sentence embeddings. To validate our results, we also evaluate twelve other SBERT models which show comparable results. Furthermore, we extend our evaluation by also considering generated definitions by the `Flan-T5-Definition` model recently fine-tuned by Giulianelli et al. (2023)[4] as this model has not been evaluated on the WiC, WSI, and LSC tasks previously.

[4] *flan-t5-definition-en-xl*

Target w	**Example** e	**Definition** d	**`LlamaDictionary`**
revitalize	This food revitalized the patient	Restore strength	Give new life or energy to
glove	Maxwell gloved his hand so that he would n't leave fingerprints , then pulled the trigger	To put a glove or gloves on .	Wear a glove to protect the hand when performing an activity

Table 9.2: Examples of pertinent definitions generated by `LlamaDictionary` for two word usages. The generated definitions are unfairly penalized by standard evaluation metrics.

9.4.1 Definition generation (DG)

> Given a target word w and an example usage e, the task is to generate a natural language definition d that is grammatical, fluent, and faithful to the meaning of the target word w as used in the example usage e (Giulianelli et al., 2020).

We assess the models in generating sense definitions for both familiar (*Seen* during training) and unfamiliar (*Unseen*) domains and styles.

For *Seen* evaluation, we use the **WordNet**, **Oxford**, and **Wiktionary** Test sets (see Table 9.1).

For *Unseen* evaluation, we consider the Test sets of two additional benchmarks comprising word usages from The **Urban** Dictionary (the largest online slang dictionary) (Ni and Wang, 2017) and **Wikipedia** (with rare words and phrases) (Ishiwatari et al., 2019). The Train set of these benchmarks were not considered during training.

		Urban	**Wikipedia**
Test	# words	25,909	56,008
	# definitions	34,974	8,193
	# def. per word	1.35	6.84

Table 9.3: Test partitions of *Unseen* DG benchmarks.

The decision to exclude **Urban** and **Wikipedia** from training was threefold. Firstly, their exclusion broadens the scope of our evaluation by considering familiar and unfamiliar usages. Secondly, it enabled a direct comparison with `Flan-T5-Definition`, a T5-based (Raffel et al., 2020) model. Finally, we refrained from fine-tuning the model with bad, slang, or offensive words, and with numerous erroneous entries (e.g., definitions comprising single Arabic numerals or part-of-speech tags) in **Urban** (Huang et al., 2021). Table 9.3 reports the main statistics of these benchmarks.

For comparison with previous work, we evaluated `LlamaDictionary` and `Flan-T5-Definition` by considering standard Natural Language Generation metrics such as BLEU (Papineni et al., 2002), NIST (Doddington, 2002), SacreBLEU (Post, 2018), ROUGE-L (Lin, 2004), METEOR (Banerjee and Lavie, 2005), and EXACT MATCH. Since some pertinent definitions may be unfairly penalized due to missing lexical overlap (see Table 9.2), we follow Giulianelli et al. (2023) and consider BERT-F1 Score (Zhang et al., 2020), which represents a semantic and thus valuable metric for this task.

9.4.2 Word-in-Context (WiC)

Given a target word w and two contexts c_1 and c_2 where w occurs, the task is to identify whether the occurrences of w in c_1 and c_2 correspond to the same meaning or not (Pilehvar and Camacho-Collados, 2019).

We evaluate the utility of sense definitions using sequence embeddings $v = m(g(w))$ on the original WiC benchmark (Pilehvar and Camacho-Collados, 2019). We refrain from using the Train set and instead generate two embeddings, v, for each context pair (one for c_1 and one for c_2) within the Dev and Test partitions (see Table 9.4). To address the WiC task, we then train a threshold-based classifier, for each tested model, using the cosine distance between the two embeddings of each pair in the Dev set. The training process involves selecting the threshold that maximizes the performance on the Dev set. Finally, we apply this classifier to conduct our evaluation over the Test set. We utilize accuracy as the assessment metric for comparison with previous work (Pilehvar and Camacho-Collados, 2019).

	WiC	
Partition	**Dev**	**Test**
# pairs	638	1,400
# words	599	1,184

Table 9.4: Test-Dev partitions for Word-in-Context.

9.4.3 Lexical Semantic Change (LSC)

Given a set of target words w and two corpora C_1 and C_2 of different time periods, the task is to rank the targets according to their degree of *lexical semantic change*[a] between C_1 and C_2 (Schlechtweg et al., 2020).

[a] "Innovations which change the lexical meaning rather than the grammatical function of a form" (Bloomfield, 1933)

We evaluate our approach on the original SemEval-English LSC benchmark (Schlechtweg et al., 2020). The dataset consists of two corpora and a test set of 46 target words (see Table 9.5). Train and Dev sets are not available as the task is set in an unsupervised scenario. To address the LSC task, we leverage popular methods generally applied using word embeddings rather than sentence embeddings (Periti and Tahmasebi, 2024a). In particular, we evaluate two different approaches:

Average Pairwise Distance (APD) is defined as *form-based* method, meaning that it quantifies change without modeling the underlying meanings of the words. Given a word w, APD computes the degree of change as the average pairwise distance between the embeddings of w generated for C_1 and C_2 (Giulianelli et al., 2020).

Average Pairwise Distance Between Sense Prototypes (APDP) is defined as *sense-based* method, meaning that it quantifies change after modeling the underlying meanings of the words via clustering. Following previous work (Rother et al., 2020) and the recent BERTopic pipeline (Grootendorst, 2022), we consider the HDBSCAN algorithm (McInnes et al., 2017). Given a word w, APDP computes the degree of change as the average pairwise distances between the sense prototypes of w in the time periods C_1 and C_2, where sense prototypes are the set of embeddings obtained by averaging the embeddings of C_1 and C_2 in each cluster, respectively (Kashleva et al., 2022).

For comparison with previous work, we utilize the Spearman rank correlation between gold scores and predictions as the assessment metric.

Test	**LSC - WSI**
# words	46
# clusters per word	9.4
max # of clusters	55
min # of clusters	1

Table 9.5: Test set for Lexical Semantic Change and Word Sense Induction, EN portion of SemEval-2020 Task 1.

9.4.4 Word Sense Induction (WSI)

Given a set of occurrences for a target word w, the task is to automatically determine the different senses of w without relying on predefined sense inventories (Agirre and Soroa, 2007).

For simplicity, we follow the recent comparison by Periti and Tahmasebi (2024a) and perform a WSI evaluation on the same benchmark used for the LSC evaluation, as it also includes gold scores for WSI. Thus, we evaluate the clustering result obtained by using HDBSCAN against labels provided for clusters in the LSC data.

As assessment metrics, we utilize Rand Index (RI) (Rand, 1971) and its Adjusted version (ARI) (Hubert and Arabie, 1985) as well as Purity (Manning, 2009). RI/ARI evaluate the similarity among two clustering results. ARI can yield low scores when a clustering result contains numerous small, yet coherent clusters. This does not necessarily indicate poor clustering quality, especially when the clusters are semantically meaningful. PUR assigns each cluster to the class that is most frequent in the cluster, measuring the accuracy of this assignment by counting the relative number of correctly assigned elements.

9.5 Evaluation results

In our evaluation, we used `Llama2Dictionary`[5] and `Llama3Dictionary`[6] with the parameters reported in Table D.2 and `Flan-T5-Definition`. See Table D.5 for specific parameters for each task.

9.5.1 Definition Generation (DG)

For the *Seen* benchmark evaluation, we consider the average performance over **WordNet** and **Oxford** (see Table 9.6). Note that, for **Wiktionary**, we do not compare with `Flan-T5-Definition` as the entire benchmark (i.e., Train-Dev-Test) has been used for training. Further details and comparisons with state-of-the-art methods across multiple benchmarks are reported in Table D.6.

For `Flan-T5-Definition`, we report the original score presented by Giulianelli et al. (2023) (reported) and the score we obtain in our evaluation (observed). We believe that slight differences, where the observed results consistently under-perform compared to the reported results, are likely due to different parameter settings (e.g., temperature or greedy decoding). Nonetheless, the results are very similar.

Compared to `Flan-T5-Definition` observed, `LlamaDictionary` obtains higher results in all considered metrics. In addition, for reported, we achieve higher results for all metrics except BERT-F1, where our result is comparable (0.889 compared to 0.909). This is an interesting result considering that `Flan-T5-Definition` has been fine-tuned on more data than `LlamaDictionary`, i.e., all Train-Dev-Test sets of **Wiktionary**.

For the *Unseen* benchmarks, previous works have typically also used the data during training and are thus not fairly comparable. We report these results in Table D.2. Thus we can evaluate only `Llama2Dictionary` and `Llama3Dictionary` and find that the latter consistently outperforms the former, unlike for the *Seen* benchmarks where the models were more even. This can be attributed to the fact that the Llama3-based model is larger than Llama2 in terms of parameters and training data.

For the *Unseen* benchmarks, the BERT-F1 scores, which rely on semantic similarity, are comparable to the *Seen* benchmarks. For the remaining scores, which rely on lexical overlap, the results for the *Unseen* benchmark are consistently, and significantly lower. We believe that this drop stems both from the issues discussed in Table 9.2 as well as the fact that the base Llama chat models, which have undergone *safety tuning*, are likely restricted from generating foul language, malicious, and toxic content that can be found in the Urban dictionary. Compared to the *Seen* benchmarks, the *Unseen* benchmarks also contain multi-word phrases for which the models have not been trained.

9.5.2 Word-in-Context (WiC)

Our results are reported in Table 9.7. Results using different SBERT models are summarized in Figure 9.2. Notably, we achieve a new state-of-the-art performance of .731 for the WiC task leveraging the definitions

[5] *Llama2Dictionary*

[6] *Llama3Dictionary*

	WordNet - Oxford *Seen*		**Urban - Wikipedia** *Unseen*	
	Llama2Dict.	Flan-T5-D. rep.	Llama2Dict.	-
	Llama3Dict.	Flan-T5-D. obs.	Llama3Dict.	Flan-T5-D. obs.
ROUGE-L	**.481**	.454	.161	-
	.400	.364	**.184**	.173
BLEU	**.402**	.257	.089	-
	.283	.266	**.100**	.095
BERT-F1	.880	**.909**	.764	-
	.889	.885	**.849**	**.849**
NIST	.938	-	.346	-
	.956	.828	**.405**	.339
SACREBLEU	**22.356**	-	4.823	-
	21.975	18.851	**5.484**	5.186
METEOR	.370	-	.151	-
	.426	.333	**.184**	.165
EX. MATCH	**50.161**	-	**.000**	-
	50.093	.110	**.000**	.000

Table 9.6: Average results for the **Definition Generation** task. The best results are highlighted in **bold**.

generated by Flan-T5-Definition + SBERT. The result by Bevilacqua et al. (2020) is particularly interesting for comparison, as it has also been obtained by relying on generated definitions. However, unlike our approach, they use multiple definitions per word usage. In contrast, we use a single definition per word usage, achieving higher results by employing both LlamaDictionary and Flan-T5-Definition.

As the WiC task requires distinguishing the underlying meaning of word occurrences, the high performance of both Flan-T5-Definition and LlamaDictionary indicates that the use of definitions is a reasonable approach to capturing the intended sense while offering interpretability.

WiC	**Accuracy**
Levine et al. (2020)	.721
Bevilacqua et al. (2020)	.711
Peters et al. (2019)	.709
Chang and Chen (2019)	.692
Flan-T5-Definition + SBERT	**.731**
Llama2Dictionary + SBERT	.729
Llama3Dictionary + SBERT	.705

Table 9.7: Evaluation results for the **Word-in-Context** task. The best result is highlighted in **bold**.

9.5.3 Lexical Semantic Change (LSC)

During our evaluation, we noticed that some of the annotated sentences present in the LSC benchmark were too long to be processed by our generative models (e.g., long word usages containing multiple sentences). This prompted us to evaluate the results by considering different sentence lengths, specifically 50, 100, 150, and 200 characters as well as the full sentence length. Our results are reported in Figure 9.3 and are consistently statistically significant. However, since we needed to discard up to 30% of sentences for LlamaDictionary, we proceeded with our experiments using up to 200 characters from each sentence.

Recent findings show that form-based approaches typically outperform sense-based approaches for the LSC task (Periti et al., 2024b) and that training models on WiC tasks enhance the modeling of lexical se-

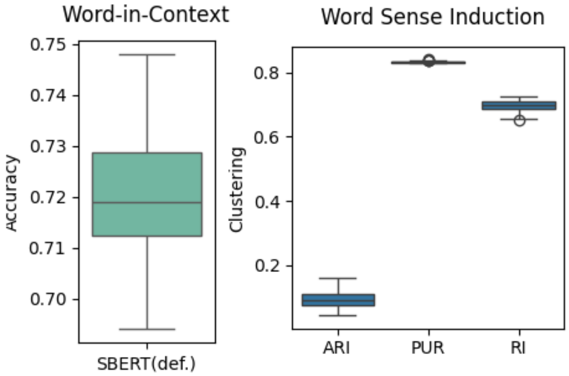


Figure 9.2: Left: Accuracy distribution on the base WiC task, using thirteen SBERT models. **Right**: ARI, PUR, and RI distribution on the WSI task, by considering our settings for the LSC task.

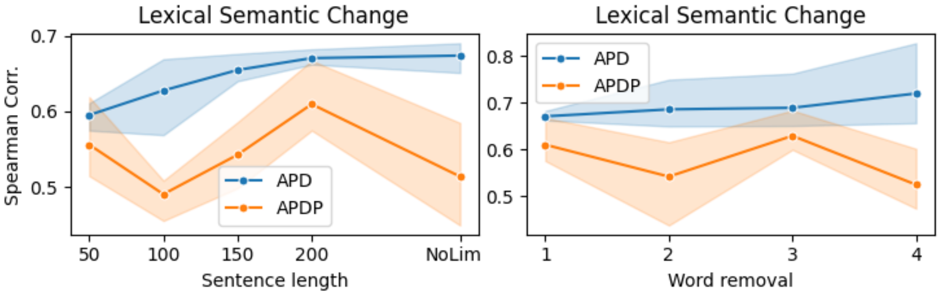


Figure 9.3: Avg. Spearman correlation by addressing LSC on different settings: different sentence length (**left**) and short word removal (**rigth**).

mantics (Arefyev et al., 2021). Similarly, we obtain higher performance for the form-based approach (APD, i.e., .662 – .682) than the sense-based one (APDP, i.e., .575 – .667), see Table 9.8. Although our results are lower than the established WiC-trained baselines, they are, on average, higher than those obtained using pretrained models (see Periti and Montanelli, 2024 for an extensive overview). Additionally, we also note that processing the generated definitions by removing short words with fewer than 2, 3 or 4 characters, in addition to punctuation, consistently boosts the performance of `Flan-T5-Definition`, reaching correlations of .755, .762 and .827, respectively (see Figure 9.3). However, we did not observe the same boost for definitions generated by `LlamaDictionary`. After reviewing a small set of generated definitions, we hypothesize that this is due to the length of definitions generated by the models, with `LlamaDictionary` trained to provide *concise* definitions (See Figure 9.1).

When compared to state-of-the-art form-based approaches, our approach achieves medium-strong correlation results but does not outperform the considered baselines. When we consider APDP, the `Llama2 Dictionary` model obtains the highest result, achieving a new state-of-the-art of .667 for interpretable LSC. This aligns with Giulianelli et al. (2023), who observe that the clusters of definitions have a lower intra-cluster dispersion compared to clusters using token and sentence embeddings.

LSC	**method**	**Spearman**
WiC-trained Aida and Bollegala (2024)	form-based	.774
WiC-trained Periti and Tahmasebi (2024a)	form-based	**.886**
Keidar et al. (2022)	form-based	.489
Giulianelli et al. (2022)	form-based	.514
`Flan-T5-Definition` + SBERT	form-based	.682
`Llama2Dictionary` + SBERT	form-based	.667
`Llama3Dictionary` + SBERT	form-based	.662
WiC-trained Periti and Tahmasebi (2024a)	sense-based	.652
Rother et al. (2020)	sense-based	.512
Montariol et al. (2021)	sense-based	.456
`Flan-T5-Definition` + SBERT	sense-based	.575
`Llama2Dictionary` + SBERT	sense-based	**.667**
`Llama3Dictionary` + SBERT	sense-based	.587

Table 9.8: Evaluation results for the **Lexical Semantic Change** task. The best result is highlighted in **bold**. Results are reported using both form-based and sense-based methods.

9.5.4 Word Sense Induction (WSI)

Our WSI evaluation relies on a recently developed benchmark originally designed for LSC. This benchmark contains cluster labels derived from manually annotated judgments of words *in-context*. These can therefore be considered as *silver* label data, rather than *gold* label data, as the clusters themselves have not been manually labeled.

Our results are reported in Table 9.9. We observe the highest results for the WiC-trained XL-LEXEME model (Cassotti et al., 2023a), and GPT-4, where the training data is unknown and thus could include both WiC data and the WSI data used in this evaluation (Balloccu et al., 2024). When compared to standard pretrained models (i.e., BERT, mBERT, XLM-R), our results are consistently higher.

In line with Periti and Tahmasebi (2024a), we observe low results in terms of ARI. We believe this stems from the quality of the original clusters to which we are comparing. The more flexible RI metric in Table 9.9 shows results comparable to the PUR scores.

In terms of the resulting clusters, we obtain an average number of clusters of 3.91 compared to the 9.61 of the original benchmark. This is in line with our intuition that definitions can be considered as prototypes of multiple word usages.

	model	**ARI**	**PUR**	**RI**
Results from Periti and Tahmasebi (2024a)	BERT	.136	.700	.629
	mBERT	.067	.644	.526
	XLM-R	.068	.737	.582
	XL-LEXEME	.273	.834	.757
	GPT-4	**.340**	**.877**	**.802**
	`Flan-T5-Definition`	.088	.832	.713
	`Llama2Dictionary`	.144	.835	.702
	`Llama3Dictionary`	.073	.832	.699

Table 9.9: Evaluation results for the **Word Sense Induction** task. The best result is highlighted in **bold**.

9.6 Discussion and considerations

Inspired by recent advancements in text generation, in this chapter, we investigated the potential of fine-tuned large language models to generate sense definitions for words *in-context*. Specifically, we fine-tuned two new Llama chat-based models, called `LlamaDictionary`, and assessed their performance along with an existing Flan-T5-Definition model on the Definition Generation task. Next, we explored their utility for modeling word meaning by addressing lexical semantic tasks such as Word-In-Context, Word Sense Induction, and Lexical Semantic Change. In our experiments, we considered the generated definitions as intermediate representations, passed through a sentence embedding model.

Our results consistently show that we can use generated definitions to explicitly model the meaning of word usages through interpretable definitions. In all tasks, the use of sentence embeddings for generated definitions outperformed the use of standard token embeddings for word occurrences, setting new state-of-the-art results. Across tasks, we find that the use of the larger 7B and 8B `LlamaDictionary` models compared to the smaller 3B T5-based model obtain slightly higher results in the Definition Generation task, while being equally strong on the lexical semantics tasks. An extension of the `LlamaDictionary` models is to fine-tune them on all the benchmarks that have been used for the `Flan-T5-Definition` model, as well as to fine-tune the models further on generated usage sentences (Malkin et al., 2021; Ma et al., 2024b).

Our evaluation using automatically generated sense definitions in this chapter paves the way for future advancements in modeling lexical semantics. For example, by offering an automatic labeling of senses, we can support the creation of lexicographic resources for all languages, including low-resource languages (Kong et al., 2022), providing a way to better know *what* change our words have experienced over time.

Limitations. In our work, we consider only English data as there are few available benchmarks on Definition Generation, neither for training nor comparison on other languages. Given the necessary resources, we believe our approach to be language-agnostic and readily applicable to other languages.

We limited our experiments to `LlamaDictionary` and `Flan-T5-Definition` due to the cost and required computational resources for fine-tuning other large language models. Such large-scale models and experimental data must be approached cautiously as they will otherwise generate enormous computational costs (both in terms of monetary and environmental costs).

A further limitation of our models arises from the fact that existing Definition Generation benchmarks occasionally include multiple definitions for the same word meanings (e.g., Table D.4). While this may serve as a form of regularization for training models, we believe that it may have influenced the uniformity in style and wording of our models. Unfortunately, statistics for these issues are non-existent. We thus advocate for further refinement to ensure consistency and coherence across definitions. We believe that, ideally, maximizing uniformity in definitions is desirable to develop models that offer consistent responses for similar word usages. This will be beneficial for any large-scale follow-up analysis relying on our evaluated approach.

In this chapter, we integrated generated definitions with sentence embeddings. However, generated definitions often display higher lexical similarity to one another compared to word usages. Given the anisotropic nature of embedding spaces in large language models (Ethayarajh, 2019), the use of sentence embeddings might complicate discerning differences in definition of different complexity for language learners (Yuan et al., 2022). We thus believe future research should also explore the utilization of definition generation models alongside more conventional text-mining methods, such as count-based models. Count-based models may offer a more straightforward approach to processing interpretable, lexical similar definitions.

Chapter 10

Modeling historical resonance

"To beer or not to beer"

Spaggiari et al., *A meta-analysis of the effects of beer consumption on cardiovascular health.* PLoS One.

10.1 Introduction

Thus far, in the preceding chapters of this thesis, we focused on the computational modeling of semantic change at word-level. Our discussion centered on lexical semantic change and modeling of word meaning by considering the temporal nature of language. In this chapter, we move our attention towards the computational modeling of semantic shift at text-level. In particular, we consider the semantic change of existing *text* (e.g., well-known phrases, sentences, multi-word expressions) that is *re-used* over time in different contexts.

As individuals, we often *reuse* someone else's words for diverse reasons and in various ways. This linguistic choice transcends cultural and temporal boundaries, representing an interesting phenomenon to study in Linguistics (Bois, 2014). For instance, linguistic scholars have investigated theories of Reception (Thompson, 1993; Hohendahl and Silberman, 1977) and Resonance (McDonnell et al., 2017; Dimock, 1997) to understand how individuals and communities interpret and reuse historical texts many years after they were written.

With the advent of digitization, recent years have seen a growing interest in computational methods for studying *text reuse*, i.e., "the reuse of existing written sources in the creation of a new text" (Clough et al., 2002). Existing methods focus on the main task of Text Reuse Detection (TRD). In TRD, text reuses are all assumed as "*topically related* to the source" (Hagen and Stein, 2011; Chiu et al., 2010), the boundaries of reused text are unknown, and the goal is to *detect* text reuse across a diachronic corpus (Seo and Croft, 2008). Whether and how the topic(s) or context(s) of a reused text differs from the source is generally overlooked. Thus, new methods are needed for modeling *recontextualization*, i.e., "the dynamic transfer-and-transformation of a text from one discourse/text-in-context to another" (Connolly, 2014; Linell, 1998).

In this paper, we propose a framework, called Topic Relatedness of Text Reuse (TRoTR), to evaluate computational methods for capturing the different recontextualizations of text reuse. In TRoTR, the boundaries of reused text are known and the goal is to distinguish reuses of the same text according to their different, latent (i.e., unlabeled) topics. As an example, consider three recontextualizations of the biblical passage *John 15:13* (in bold):

(1) It's the wonderful pride month!! ❤️🧡💛💚💙💜 Honestly pride is everyday! Love is love don't forget I love you ❤️. Remember this! John 15:12-13: "My command is this: Love each other as I have loved you. **Greater love has no one than this: to lay down one's life for one's friends**"

(2) At a large Crimean event today Putin quoted the Bible to defend the special military operation in Ukraine which has killed thousands and displaced millions. His words "**There is no greater love than if someone gives soul for their friends**". And people were cheering him. Madness!!!

(3) "Freeing people from genocide is the reason, motive & goal of the military operation we started in the Donbas & Ukraine", Putin says, then quotes the Bible: "**There is no greater love than to lay down one's life for one's friends.**" It's like Billy Graham meets North Korea

In this example, the biblical passage is incorporated within three texts with different topic recontextualizations. In particular, the text (1) has a different topic with respect to text (2) and (3), while the texts (2) and (3) are topic related. In TRoTR, we support the recognition of such a kind of recontextualizations by leveraging the notion of topic relatedness. TRoTR represents a new opportunity in Natural Language Processing (NLP) and can be used to distinguish recontextualizations of any kind of text reuse (e.g., proverbs, Ghosh and Srivastava, 2022), to investigate phenomena such as the use of misquotations (Porrino et al., 2008) and dogwhistles (Hertzberg et al., 2022), as well as to provide in-context interpretation to vague utterances, with special focus on enhancing the LLMs' capabilities to this end (DeVault and Stone, 2004).

Chapter outline.

This chapter includes materials originally published in the following publication:

> Francesco Periti, Pierluigi Cassotti, Stefano Montanelli, Nina Tahmasebi, and Dominik Schlechtweg. 2024**c**. TRoTR: A Framework for Evaluating the Re-contextualization of Text Reuse. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, Florida, USA. Association for Computational Linguistics.

In this chapter, we introduce a novel framework, called TRoTR, with two NLP tasks called Text Reuse in-Context (TRiC) and Topic variation Ranking across Corpus (TRaC). The chapter is organized as follows. In Section 10.2, we frame our framework within the relevant literature. In Section 10.3, we present the TRoTR framework and outline the structure of TRiC and TRaC. In Section 10.4, we present the TRoTR benchmark containing gold labels derived by human judgments of topic relatedness in context pairs. The judgments show an inter-annotator agreement of .811, calculated by the average pairwise correlation on assigned assessments. In Section 10.5, we describe the setup of our evaluation. In Section 10.6, we present the results of our experiments. In particular, we evaluate 36 SBERT models by considering 4 settings. Our results reveal that these models reach high performance (correlation 0.6-0.8), but are more sensitive to semantic similarity rather than topic relatedness. Finally, we summarize the findings of this chapter, as well as its main limitations, in Section 10.7.

10.2 Background and related work

Works related to TRoTR are about text reuse and recontextualization, semantic textual similarity and relatedness, and topic modeling and annotation.

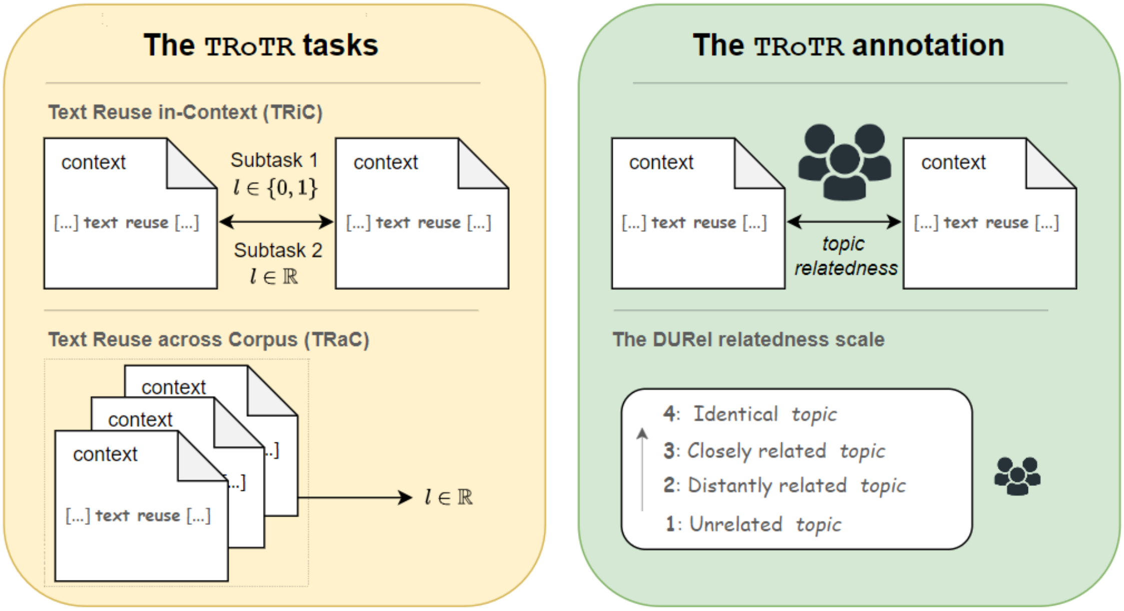


Figure 10.1: The TRoTR framework consists of two tasks, called Text Reuse in-Context (TRiC) and Topic variation Ranking across Corpus (TRaC), along with a corresponding annotation process. We use [...] to denote the left and right context of a target text-reuse excerpt.

Text reuse and recontextualization. Although multiple facets of text reuse have been investigated, such as historical (Büchler et al., 2014), cross-lingual (Muneer and Nawab, 2022), allusive (Manjavacas et al., 2019), explicit (Franzini et al., 2018), non-literal (Moritz et al., 2016), and local (Seo and Croft, 2008), computational approaches primarily focuses on *detecting* instances of text reuse. To the best of our knowledge, studies extending beyond mere TRD often leverage text metadata to analyze reuse within temporal and spatial graphs (Khritankov et al., 2015; Smith et al., 2013; Xu et al., 2014). However, these studies do not specifically focus on capturing how the reused text is recontextualized, thereby leaving a gap in the current literature.

Among recent advancements in NLP, some works are related to the recontextualization of text. Wilner et al. (2021) focus on Narrative Analysis by investigating how the recontextualization of events across whole stories impacts word embeddings. Ghosh and Srivastava (2022) introduce a benchmark for evaluating the LLMs' capability of generating proverbs in-context of narratives.

Over the past few years, there has been growing interest in quotations, i.e. "well known phrases or sentences that we use for various purposes such as emphasis, elaboration, and humor" (Lee et al., 2016). This interest extends to various forms of quotations spanning from epigraphs (Bond and Matthews, 2018) to biblical references (Moritz et al., 2016). In particular, there has been a surge of attention in recommendation systems that offer off-the-shelf quotations based on provided context (Wang et al., 2023, 2022, 2021b).

Text 1	**Text 2**	**Semantic Textual Similarity**	**Semantic Textual Relatedness**	**Semantic Textual *Topic* Relatedness**
It's the wonderful pride month!! ❤️🧡💛💚💙💜 Honestly pride is everyday! Love is love don't forget I love you ❤️. Remember this! John 15:12-13: "My command is this: Love each other as I have loved you. **Greater love has no one than this: to lay down one's life for one's friends**"	Happy Pride Month! ❤️ Remember, pride isn't just for a month—it's a daily celebration! Love knows no boundaries, and I want you to know that I cherish you every single day. ❤️ Let's always remember these powerful words from John 15:12-13: "My command is this: Love each other as I have loved you. **Greater love has no one than this: to lay down one's life for one's friends**"	✓ paraphrase	✓ related in some aspects	✓ related in topic
"Freeing people from genocide is the reason, motive & goal of the military operation we started in the Donbas & Ukraine", Putin says, then quotes the Bible: "**There is no greater love than to lay down one's life for one's friends.**" It's like Billy Graham meets North Korea	At a large Crimean event today Putin quoted the Bible to defend the special military operation in Ukraine which has killed thousands and displaced millions. His words "**There is no greater love than if someone gives soul for their friends**". And people were cheering him. Madness!!!	× neither paraphrases nor entailment	✓ related in some aspects	✓ related in topic
It's the wonderful pride month!! ❤️🧡💛💚💙💜 Honestly pride is everyday! Love is love don't forget I love you ❤️. Remember this! John 15:12-13: "My command is this: Love each other as I have loved you. **Greater love has no one than this: to lay down one's life for one's friends**"	At a large Crimean event today Putin quoted the Bible to defend the special military operation in Ukraine which has killed thousands and displaced millions. His words "**There is no greater love than if someone gives soul for their friends**". And people were cheering him. Madness!!!	× neither paraphrases nor entailment	✓ related in some aspects	× unrelated in topic
You are altogether beautiful, my darling; there is no flaw in you. Charm is deceitful, and beauty is vain, but a woman who fears the Lord is to be praised	At a large Crimean event today Putin quoted the Bible to defend the special military operation in Ukraine which has killed thousands and displaced millions. His words "**There is no greater love than if someone gives soul for their friends**". And people were cheering him. Madness!!!	× neither paraphrases nor entailment	× unrelated in any aspects	× unrelated in topic

Table 10.1: Examples of *semantic textual similarity*, *semantic textual relatedness*, and *topic relatedness*. The first and last pair of sentences are examples of paraphrases and semantically unrelated content, respectively. Most people will agree that the second pair of sentences is more related in topic than the third pair of sentences. However, some people may still consider the third pair as semantically related due to the presence of the same quotation.

Semantic textual similarity and relatedness. In NLP, a possible option for assessing text recontextualization is to use *semantic* (textual) *similarity*. However, semantic similarity is traditionally used as a metric to assess paraphrases or entailment equivalence between two texts (Hercig and Kral, 2021; Konopík et al., 2017; Cer et al., 2017; Agirre et al., 2016, 2015, 2014, 2013, 2012); thus, it is not suitable for TRoTR. *Semantic* (textual) *relatedness* has been long recognized as a core aspect in understanding the meaning of texts (Miller and Charles, 1991), and encompasses a multitude of intricate relationships, such as sharing a common *topic*, expressing similar viewpoints, or originating from the same temporal period (Abdalla et al., 2023). However, there is no universally accepted linguistic theory or set of guidelines for evaluating relatedness. Its assessment is inherently more complex than semantic similarity, as two texts may lack semantic similarity but still be semantically related through some textual relationship (see Table 10.1).

Topic modeling and annotation. An alternative method for assessing text recontextualization is by analyzing topics where text is reused (Jin and Spence, 2021; Kim et al., 2018). Topic models can be useful tools to discover latent topics in collections of documents (Abdelrazek et al., 2023), either as probability distributions like LDA (Blei et al., 2003) or clustering of embeddings like BERTopic (Grootendorst, 2022). When applied, the derived topics need to be carefully evaluated against benchmarks containing manually derived ground truth. As topics represent vague concepts, different guidelines for deriving ground truth use different topic definitions tailored to the specific interests of analysis (Orita et al., 2014). Generally, these guidelines result in manual annotations of topic labels that typically differ across annotators and thus require post-processing techniques to be uniform and standardized (Poursabzi-Sangdeh and Boyd-Graber, 2015). For example, annotators can use different wording to express the same concept.

As a result, there is no well-established guideline for annotating topics. However, common to different guidelines is a definition of topic that relies on the notion *what the text is about* (Bauwelinck and Lefever, 2020; Hovy and Lin, 1998).

10.3 The TRoTR framework

The TRoTR framework consists of two tasks, called Text Reuse in-Context (TRiC) and Topic variation Ranking across Corpus (TRaC), along with a corresponding annotation process (see Figure 10.1). TRiC and TRaC are grounded on human judgments of a specific facet of semantic relatedness (see Section 10.2) that considers the extent to which two texts share a common *topic*. We call this facet **topic relatedness** (see Table 10.1 for an example). In our study, the definition of topic follows the popular notion of *what the text is about*.

When dealing with complex problems, such as recontextualization, a general approach involves starting with a smaller sub-problem to establish a focused foundation before further expanding. Thus, we first present TRiC as a *context-pair level* task. Then, we present TRaC as a more complex *corpus-level* task that must be addressed to identify potential varying targets for real, in-depth analysis.

10.3.1 Tasks

In the TRoTR tasks, instances of text reuse are presented within different contexts, each representing a new recontextualization of the original text.

Text Reuse in-Context frames a text reuse t within two different contexts c_1 and c_2. The goal is to assess the topic relatedness of c_1 and c_2. TRiC includes two subtasks, namely *binary classification* and *ranking*. These subtasks resemble the structure of the Word-in-Context task (Pilehvar and Camacho-Collados, 2019) and the Graded Word Similarity in Context task (Armendariz et al., 2020b), respectively. However, while they focus on distinguishing the different meanings words can have in different contexts, TRiC focuses on distinguishing different topics in which text is reused.

Each TRiC instance is associated with a binary label $l \in \{0, 1\}$ and a continuous score $1 \leq s \leq 4$.

- Subtask 1 - *binary classification*: the task is to identify, for each instance, whether the contexts c_1 and c_2 share roughly the same topic (i.e., $l = 1$) or not (i.e., $l = 0$).
- Subtask 2 - *ranking*: the task is to rank the TRiC instances according to the degree of topic relatedness s of the contexts c_1 and c_2.

Topic variation Ranking across Corpus frames a text reuse t within a corpus C that includes various contexts c_i where t occurs. TRaC resembles the structure of the Lexical Semantic Change (LSC) detection task defined by (Schlechtweg et al., 2018; Kutuzov and Pivovarova, 2021b). However, while this focuses on assessing the semantic change of a word, TRaC focuses on assessing the *topic variation* of a reused text. Each TRaC instance is associated with a continuous score $s \in [0, 1]$ of topic variation that indicates the variability in topic usages for a target text reuse t across the corpus C. Specifically, a score of 1 indicates that a target is associated with a high number of topics, while a score of 0 indicates that a target is associated with a single topic.

Given a set of target text reuses $t \in T$, the task is to rank the text reuses by the degree of topic variation across the corpus C.

10.3.2 Annotation process

The TRoTR annotation process is enforced to collect human judgments of topic relatedness (see Table 10.1). In our study, we sidestep the need for annotating topics explicitly using a well-established paradigm adopted for modeling word meaning. Our intuition is that annotating topic relatedness, instead of relying on explicit topic labels, closely mirroring recent work exemplified in the Word-in-Context task (Pilehvar and Camacho-Collados, 2019), which relies on annotating meaning relatedness rather than explicit sense labels.

Annotators are asked to evaluate the *topic relatedness* of different text reuse instances $\langle t, c_1, c_2 \rangle$, where t is a target text reuse, and c_1 and c_2 are two different contexts in which t occurs.

The topic relatedness is evaluated by utilizing the four-point DURel relatedness scale (Schlechtweg et al., 2024), with annotators following instructions inspired by the guidelines from Erk et al. (2013), as well as those provided for SemEval-2020 Task 1 (Schlechtweg et al., 2020) and the PLATOS project (Bauwelinck and Lefever, 2020). [1]

10.4 The TRoTR benchmark

The TRoTR benchmark is composed of human-annotated instances of text reuse. Specifically, we first manually collected and curated tweets containing text reuse instances. We then incorporated gold labels derived by human annotations.

10.4.1 Data

Inspired by Moritz et al. (2016); Büchler et al. (2014), we focus on text reuse of biblical passages because they typically show high context variety (Greenough, 2021; Cheong, 2014), the degree of which we aim to study. Moreover, they are frequently and explicitly mentioned *in-context*, often with an identifying reference (e.g., *John 15:13*). Tweets were collected through a manual search process, thus allowing us to avoid a TRD phase and its validation.

For a set of 42 target passages we collected 30 tweets each. These were curated by experts by removing minor word variations in phrasing that can stem from the use of e.g., different Bible versions.

10.4.2 Human judgments

We collected judgments according to the procedure outlined in Section 10.3.2. Specifically, we recruited four native English speakers as annotators. Annotators were trained and tested on a small set of instances in an online tutorial.

For each target passage t, we generate all possible context pairs where the contexts are chosen from the 30 tweets. We then randomly sampled 150 context pairs. These were presented to annotators in randomized order to be judged for topic relatedness. Each context pair received at least two judgments, although the majority received three.

The outcome of our annotation pipeline is a dataset of 6,300 annotated context pairs. We measured inter-annotator agreement on judgments using Krippendorff's α coefficient (Krippendorff, 2019) and the weighted mean of Spearman correlations (Spearman, 1987) between annotator pairs. Similar to previous studies that reported Krippendorff's α of .439 (Loureiro et al., 2022) and weighted mean of Spearman correlation between annotator judgments ranging from .550 to .680 (Erk et al., 2013; Schlechtweg et al., 2018), we obtained a comparable Krippendorff's α score of .420 and Spearman correlation of .506.

[1] The annotation guidelines for TRoTR, along with its benchmark, and our code, are submitted and will be publicly available.

10.4.3 Deriving gold labels

Following Loureiro et al. (2022), we employ filtering criteria for the annotation instances to reduce uncertainty and ensure a more controlled setting.

For TRiC, we first filtered out all instances with high disagreement[2], e.g. an instance with three different judgments where it is unclear which the gold label could be. We also enforce a clear-cut separation by filtering out all the instances where the average judgment score is between 2 and 3. This filtering results in a more refined dataset of 3,821 annotated context pairs, characterized by a Krippendorff's α agreement of .709 and a weighted average pairwise Spearman agreement of .811.

For TRaC, we adopted a different filtering approach at the level of targets to ensure a comparable number of instance pairs when deriving the gold labels. Specifically, we filtered out the targets t where the weighted average pairwise Spearman agreement is below .150 leading to the exclusion of 2 targets.

TRiC labels. For each instance, we aggregate the judgments of all annotators by averaging. We then directly use the average judgment s of each instance to derive binary labels and continuous scores for Subtask 1 and Subtask 2.

For Subtask 1, we binarize s as 1 if $s \geq 2.5$ or as 0 if $s < 2.5$ and associate each instance with the corresponding binary label. A threshold of 2.5 is a midpoint split on the judgment scale. It follows that the 0 label consists of Unrelated and Distantly related annotations, while label 1 consists of Identical and Closely related annotations. Overall, our benchmark includes a total of 2,621 examples with label 0 and a total of 1,200 examples with label 1.

For Subtask 2, we directly utilize the continuous score s for each instance.

TRaC labels. For each target, we use a judgment summary measure similar to the DURel EARLIER/LATER measures introduced by Schlechtweg et al. (2018) in the field of LSC (Periti and Montanelli, 2024; Tahmasebi et al., 2021a). This involves computing the average of annotator judgments over all instances for a target. Lower scores correspond to greater topic variation, while greater scores (i.e., more Identical annotations) are associated with less topic variation.

10.5 Evaluation setup

We use the TRoTR tasks and benchmarks to evaluate the ability of sequence-level models to capture topic relatedness and variation in different text recontextualizations to set baselines for the tasks.

Because Sentence-BERT (SBERT) models are recognized to be the state-of-the-art architecture for addressing sequence-level tasks (Reimers and Gurevych, 2019), we choose a range of different SBERT models tailored for sequence-level embeddings and textual similarity.

[2] We consider high disagreement to be a difference between the maximum and the minimum judgment of 2 or 3.

10.5.1 SBERT models

We consider 36 SBERT models trained on a wide range of tasks including Paraphrasis, Semantic Similarity, and Question Answering. Specifically, we evaluate all the (non-image based) pre-trained models available at `https://www.sbert.net/index.html`. We evaluate each SBERT model in its pre-trained version (base) and three different settings, namely:

- *+MASK*: given an instance $\langle t, c_1, c_2 \rangle$, we mask the text-reuse excerpt t in the contexts c_1 and c_2 to prevent that the topic estimate of topic relatedness is influenced by the common t in c_1 and c_2. To this end, we replace t in c_1 and c_2 with a dash (i.e., "-");
- *+FT*: we fine-tune the pre-trained model on TRiC instances using the *contrastive loss* (Hadsell et al., 2006). This loss minimizes the distance between embeddings of similar sentences and maximizes the distance for dissimilar sentences;
- *+FT+MASK*: we combine both the +FT and +MASK settings, meaning that we fine-tune the model and then evaluate it by considering contexts where targets are masked.

SBERT architectures. Each SBERT model has been pre-trained using one of two architectures:

- *Bi-Encoder* models are designed to produce a sequence embedding for an input text sequence. Given an instance $\langle t, c_1, c_2 \rangle$, we independently feed a Bi-Encoder model with the sequence c_1 and c_2 to obtain the corresponding sequence embeddings u and v. Similar to Abdalla et al. (2023), we use the cosine similarity between u and v as an estimate of the topic relatedness between c_1 and c_2.
- *Cross-Encoder* models are designed to produce an output value that indicates the similarity of two input sequences. Thus, given an instance $\langle t, c_1, c_2 \rangle$, we simultaneously pass the sequences c_1 and c_2 to the Cross-Encoder model and use the output value as an estimate of the topic relatedness between c_1 and c_2.

10.5.2 TRiC evaluation

Similar to the WiC tasks (e.g., Pilehvar and Camacho-Collados, 2019), we split the `TRoTR` benchmark into three distinct partitions, namely training set (Train), development set (Dev), and test set (Test), comprising approximately 80%, 10%, and 10% of the instances, respectively. To strengthen the robustness of the evaluation, ten randomized Train-Dev-Test splits were generated (see Appendix E.1). The average performance across all the splits is used as reference for comparison.

Additionally, inspired by Raganato et al. (2020), we include the evaluation of target text reuse t that are unseen during fine-tuning. The goal is to evaluate the ability of models to generalize the assessment of topic relatedness. Specifically, we fine-tune each considered model on the Train set and we evaluate it on two different Test sets: i) the standard Test set, containing instances $\langle t, c_1, c_2 \rangle$ whose target t was either seen or unseen during fine-tuning; and ii) the **Out-of-Vocabulary** (OOV) Test set, containing only instances

$\langle t, c_1, c_2 \rangle$ whose target t was not seen during fine-tuning. OOV Test set represents half of the Standard Test set.

For TRiC Subtask 1, we need to define a threshold to determine instances $\langle t, c_1, c_2 \rangle$ where c_1 and c_2 share roughly the same topic or not. Thus, given a model, we tune a threshold-based classifier on the Dev set. Specifically, for each instance $\langle t, c_1, c_2 \rangle$ in Dev, we use the model to predict the topic relatedness between c_1 and c_2. Then, we determine the optimal threshold that maximized the Weighted F1 (Harbecke et al., 2022) score over the Dev set. Finally, we apply this threshold to both the Train and Test sets. Due to the unbalanced distribution of gold binary labels, we evaluate models using the F1 metric. Precision (PR) and Recall (RE) for each individual class are also reported for completeness.

For TRiC Subtask 2, given a model, we directly use its predictions as estimates of topic relatedness. Then, we evaluate the model using Spearman correlation (SP) with continuous gold scores.

10.5.3 TRaC evaluation

Similar to the LSC tasks (e.g., Schlechtweg et al., 2020), we consider an *unsupervised* scenario. In particular, motivated by the limited number of targets (i.e., 42), we do not split the benchmark into Train-Dev-Test partitions with the aim to mitigate the potential evaluation impact of a small Test set. Without training instances, the configurations with +FT and +FT+MASK are not applicable to TRaC.

To quantify the topic variation of a target, we adopted the same approach used for determining the gold scores. Thus, given a model, the topic variation of a target t is calculated as the average prediction of topic relatedness across all the annotated $\langle t, c_1, c_2 \rangle$ pairs. We then evaluate models using Spearman correlation (SP) with gold scores.

10.6 Evaluation results

First, we evaluated an extensive set of pre-trained SBERT models on the TRiC task (see Table E.2 in Appendix). Then, for simplicity, we opted to consider and fine-tune a smaller set of models, precisely the top-five models by SP over the Train sets. Since we did not perform any training over the models, the Train sets act as a larger set for testing the models. Specifically, we chose: *all-distilroberta-v1* **(ADR)**, *distiluse-base-multilingual-cased-v1* **(DBM)**, *paraphrase-multilingual-MiniLM-L12-v2* **(PAM)**, *paraphrase-multilingual-mpnet-base-v2* **(PAR)**, and *multi-qa-mpnet-base-cos-v1* **(MQA)**. In particular, ADR and DBM are Bi-Encoders for English. PAM and PAR are multilingual Bi-Encoders fine-tuned on paraphrase pairs. Similarly, MQA is a multilingual Bi-Encoder fine-tuned on question-answer pairs.

As a general remark on our initial evaluation, we note that Bi-Encoder models consistently exhibit superior performance compared to Cross-Encoder models in both TRiC Subtask 1 and Subtask 2. This finding aligns with the recent comparisons by Ishihara and Shirai (2022) and Cassotti et al. (2023a) for News Article

	Standard Test set								**Out-of-vocabulary (OOV) Test set**							
	Label 0			*Label 1*			***All***		*Label 0*			*Label 1*			***All***	
Models	PR	RE	F1	PR	RE	F1	F1	SP	PR	RE	F1	PR	RE	F1	F1	SP
ADR	**.95±.03**	.47±.13	.62±.11	.42±.11	.93±.04	.57±.10	.61±.10	.55±.09	**.94±.07**	.45±.20	.58±.20	.38±.19	**.93±.06**	.51±.18	.58±.16	.48±.20
+FT	**.95±.03**	**.61±.15**	**.73±.11**	**.50±.14**	**.93±.03**	**.64±.10**	**.71±.10**	**.66±.07**	.91±.12	**.49±.24**	**.61±.22**	**.40±.21**	.91±.06	**.52±.18**	**.61±.18**	**.51±.22**
+MASK	*.89±.05*	*.87±.07*	*.87±.03*	*.70±.14*	*.72±.12*	*.69±.07*	*.82±.03*	*.67±.06*	*.90±.07*	*.85±.10*	*.87±.05*	*.62±.21*	*.71±.18*	*.63±.14*	*.82±.05*	*.62±.15*
+FT+MASK	*.90±.07*	*.89±.07*	*.89±.03*	*.75±.12*	*.76±.12*	*.74±.05*	*.85±.04*	*.71±.05*	*.87±.11*	*.88±.09*	*.87±.06*	*.66±.20*	*.70±.15*	*.65±.09*	*.82±.06*	*.63±.15*
DBM	.96±.02	.26±.12	.40±.14	.35±.09	**.97±.03**	.51±.09	.43±.12	.54±.09	**.96±.08**	.21±.19	.31±.23	.31±.14	**.97±.05**	.45±.16	.38±.18	.44±.23
+FT	**.97±.02**	**.46±.17**	**.60±.15**	**.43±.10**	.96±.03	**.58±.09**	**.61±.13**	**.64±.07**	.93±.15	**.34±.23**	**.46±.26**	**.34±.14**	.95±.05	**.49±.15**	**.50±.19**	**.48±.29**
+MASK	*.87±.07*	*.88±.07*	*.87±.03*	*.72±.14*	*.66±.16*	*.66±.09*	*.81±.03*	*.64±.04*	*.88±.09*	*.88±.09*	*.87±.05*	*.66±.23*	*.64±.25*	*.58±.19*	*.82±.04*	*.58±.12*
+FT+MASK	*.88±.06*	*.89±.07*	*.88±.04*	*.74±.11*	*.70±.13*	*.70±.04*	*.83±.03*	*.66±.04*	*.85±.12*	*.87±.09*	*.85±.08*	*.63±.19*	*.58±.20*	*.57±.13*	*.80±.08*	*.58±.14*
PAM	**.96±.02**	.46±.09	.61±.08	.41±.09	**.96±.02**	.57±.08	.61±.07	.58±.08	**.96±.04**	.43±.17	**.57±.16**	**.37±.15**	**.95±.05**	**.52±.15**	**.59±.12**	.49±.22
+FT	.95±.03	**.59±.12**	**.72±.11**	**.48±.09**	.92±.04	**.63±.08**	**.70±.09**	**.66±.06**	.90±.18	**.45±.21**	**.57±.23**	**.37±.13**	.92±.06	.51±.13	**.59±.17**	**.51±.22**
+MASK	*.89±.05*	*.88±.06*	*.88±.03*	*.71±.10*	*.72±.10*	*.70±.05*	*.83±.03*	*.67±.04*	*.89±.09*	*.86±.09*	*.87±.06*	*.65±.19*	*.71±.18*	*.65±.12*	*.83±.05*	*.60±.13*
+FT+MASK	*.90±.05*	*.90±.03*	*.90±.03*	*.76±.07*	*.77±.06*	*.76±.03*	*.86±.03*	*.69±.04*	*.88±.10*	*.89±.05*	*.88±.06*	*.68±.13*	*.73±.11*	*.69±.07*	*.84±.06*	*.60±.12*
PAR	**.95±.03**	.40±.10	.56±.09	.39±.09	**.95±.04**	.55±.08	.56±.07	.56±.09	**.93±.11**	.35±.18	.49±.19	.34±.15	**.95±.06**	.49±.16	.52±.15	.47±.25
+FT	**.95±.05**	**.60±.10**	**.73±.08**	**.49±.10**	.93±.05	**.63±.08**	**.71±.07**	**.66±.06**	.91±.17	**.46±.21**	**.58±.21**	**.38±.16**	.91±.08	**.51±.15**	**.59±.18**	**.53±.24**
+MASK	*.89±.05*	*.85±.07*	*.87±.04*	*.69±.10*	*.75±.11*	*.70±.05*	*.83±.03*	*.68±.03*	*.90±.08*	*.83±.13*	*.86±.07*	*.63±.19*	*.75±.17*	*.65±.10*	*.82±.05*	*.62±.11*
+FT+MASK	*.89±.06*	*.91±.05*	*.90±.03*	*.78±.09*	*.73±.11*	*.74±.05*	*.86±.03*	*.70±.04*	*.87±.11*	*.90±.07*	*.88±.06*	*.68±.16*	*.66±.18*	*.64±.11*	*.83±.07*	*.61±.14*
MQA	.94±.03	.42±.11	.58±.11	.40±.10	**.94±.03**	.55±.09	.58±.09	.55±.09	**.94±.09**	.39±.19	.53±.20	.36±.19	**.96±.03**	.50±.18	.55±.16	.49±.21
+FT	**.96±.03**	**.61±.13**	**.74±.10**	**.50±.10**	**.94±.04**	**.65±.08**	**.72±.09**	**.68±.06**	.92±.15	**.47±.22**	**.60±.24**	**.39±.16**	.94±.05	**.53±.15**	**.61±.19**	**.54±.21**
+MASK	*.88±.05*	*.87±.07*	*.88±.04*	*.71±.10*	*.71±.12*	*.69±.06*	*.83±.04*	*.68±.05*	*.89±.07*	*.86±.10*	*.87±.06*	*.63±.18*	*.69±.16*	*.63±.13*	*.83±.05*	*.62±.13*
+FT+MASK	*.90±.05*	*.91±.04*	*.90±.03*	*.77±.08*	*.76±.09*	*.76±.05*	*.86±.03*	*.72±.04*	*.88±.10*	*.90±.04*	*.88±.06*	*.67±.16*	*.69±.16*	*.65±.11*	*.84±.06*	*.63±.13*

Table 10.2: TRiC evaluation on Subtask 1 and Subtask 2 for both Test and OOV Test sets. For Subtask 1, precision (PR), recall (RE), and Weighted -F1 scores (F1) are reported for both label 0 (i.e., different topics) and label 1 (i.e., roughly identical topics). For Subtask 2, Spearman correlation (SP) is reported on the overall set of instances. Standard deviations (±) across the 10 Test splits are presented for comparative analysis. For each metric, the best performance of the comparison between pre-trained/fine-tuned models is highlighted in **bold**. Results for masking settings are reported in *italic*.

Models	ADR	DBM	PAM	PAR	MQA
	+MASK	*+MASK*	*+MASK*	*+MASK*	*+MASK*
Spearman	.72	.66	.66	.73	.65
	.84	*.80*	*.81*	*.76*	*.80*

Table 10.3: TRaC evaluation using the pre-trained models alone and in the +MASK setting (*italic*).

Similarity and LSC, challenging the idea that the use of cross-attention benefits Cross-Encoder architectures in sequence-level tasks (Lee et al., 2023; Thakur et al., 2021). In the following, we first present the results of our evaluation by comparing the use of pre-trained and fine-tuned models (+FT); then, we discuss the results in the masking settings (+MASK, +FT+MASK). We report in Table 10.2 and 10.3 the overall results for TRiC and TRaC, respectively.

10.6.1 TRiC: pre-trained vs. fine-tuned

Across the overall *standard* Test sets, when *pre-trained* models are used for Subtask 1, we observe high precision (PR) values, ranging from .93 to .96, and low recall (RE) values ranging from .21 to .47 for label 0 (i.e., different topics). Conversely, for label 1 (i.e., roughly identical topics), we observe an inverse trend of performance, with PR values ranging from .31 to .42 and RE values ranging from .93 to .97. Such results suggest that SBERT models face difficulties in distinguishing different recontextualization. For Subtask 1, we observe a moderate F1-score (F1) ranging from .43 to .61; for Subtask 2, we observe only moderate Spearman correlation coefficients (SP) ranging from .54 to .58.

Additional results for the *OOV* Test sets are reported in Table 10.2. We note that the results for the OOV

Test sets are lower in performance while being associated to higher standard deviations. For pre-trained models, we attributed this drop to (1) the unbalanced number of instances and labels available for each target; (2) that the inter-annotator agreements differ between targets. If target words with a small number of instances or lower inter-annotator agreement fall in the OOV Test sets, then the performance will be much lower. Finally, (3) the size of the OOV Test sets is smaller because it splits the standard Test sets in two halves.

Fine-tuning: When the pre-trained models are *fine-tuned* on TRiC instances (i.e., +FT), we observe a significant improvement in performance for both Subtask 1 and Subtask 2 on both the standard Test set and the OOV Test set. This observation indicates that fine-tuning SBERT models on TRiC instances enhances their capability to contextualize a sequence *in-context*. In particular, the improvement is more pronounced on the standard Test sets than on the OOV Test sets. We attribute this discrepancy to the limited size of our benchmark that includes a small number of target quotations sufficient for testing purposes. A larger number of targets will further improve the models' generalization capability. For Subtask 1, we observe a F1 ranging from **.61** to **.72** (standard) and from **.50** to **.61** (OOV); for Subtask 2, we observe SP coefficients ranging from **.64** to **.68** (standard) and **.51** to **.54** (OOV).

10.6.2 TRiC and TRaC: masking settings

When pre-trained and fine-tuned models are used in the masking settings (i.e., +MASK and +FT+MASK), we observe a significant improvement in performance for both TRiC and TRaC. Notably, this improvement for TRiC is substantially larger compared to the one observed in the prior comparison (pre-trained vs. fine-tuned), with +FT+MASK exhibiting slightly superior performance to +MASK. We attribute this improvement to the fact that, in the masking settings, models are compelled to pay more attention to the surrounding contexts of reused texts, thereby fostering a more comprehensive understanding of topic relatedness.

For TRiC, we observe the following performance. For Subtasks 1, we observe a F1 ranging from .81 to .83 and from **.82** to **.86** for +MASK and +FT+MASK, respectively. For Subtask 2, we observe a SP coefficients ranging from .60 to .68 and from **.60** to **.72** for +MASK and +FT+MASK, respectively.

For TRaC, we observe SP coefficients ranging from **.65** to **.73**. Conversely, when pre-trained models are used in the +MASK setting, SP coefficients exhibit a substantial improvement, ranging from **.76** to **.84**.

10.6.3 Discussion

The results found in our experiments underscore the difficulty of SBERT models in distinguishing different text recontextualizations. This, despite the fact that SBERT models are the state-of-the-art for sequence-level tasks. As a matter of fact, pre-trained models exhibit a bias toward their typical pre-training focus, namely *semantic similarity*, while demonstrating only a superficial understanding of *topic relatedness*. Although the masking settings seem to offer a valuable workaround to sidestep the problem, we claim that their use is generally undesirable in real scenarios involving text reuse. First, because masking may disrupt the natural flow of sentences precluding to obtain optimal performance. Second, because the boundaries of text reuse are often nuanced or unbalanced in different recontextualizations, when considering a form of text reuse broader than explicit quotation that implicitly reuses text *in-context*. In such cases, masking may result in the removal of crucial contextual information.

Consequently, to provide a more accurate modeling of text-reuse *in-context*, we argue that there is a clear imperative to develop or fine-tune novel models specifically tailored on topic relatedness. In this regard, `TRoTR` represents a valuable framework for evaluating language models that extend existing benchmarks on sentence-pair regression tasks, such as Semantic Textual Similarity (Agirre et al., 2012) and Semantic Textual Relatedness (Abdalla et al., 2023). While current benchmarks rely on a notion of *similarity* or *relatedness*, they overlook the potential impact of shared substrings, such as text-reuse excerpts, on computational estimates.

10.7 Discussion and considerations

To the best of our knowledge, this work represents a first pioneering effort in the computational modeling of *recontextualization*. We relied on the notion of *topic relatedness* to introduce a novel framework named Topic Relatedness of Text Reuse (`TRoTR`) with two tasks: Text Reuse in-Context (TRiC) and Topic variation Ranking across Corpus (TRaC). The tasks are inherently difficult as topic relatedness is under-defined, and under-researched, therefore this paper presents important steps forward.

First, we presented a human-annotated benchmark of text reuse instances extracted from Twitter. This benchmark can be used to support Linguistic Recycling and Reception studies, ranging from misuse and dog whistles to the study of author influence. Using the framework, the benchmark can easily be extended in future work to cover more diverse sets of text reuse from other sources, e.g., literature and political text.

Next, we comprehensively evaluate SBERT models on the TRiC and TRaC tasks. We find that the Bi-Encoder models outperform the Cross-Encoder models. Additionally, we evaluate the considered models by masking the occurrences of text reuse and find that the models exhibit a greater sensitivity to semantic similarity rather than topic relatedness. These results now constitute a *baseline* for continued research and can be used as a comparison for improved models and architectures.

Future work. Text reuse is inherently *diachronic* and can take place both over short and long time spans. The `TRoTR` framework is applicable to address the recontextualization problem across time, space, or domain. In our ongoing work, we will extend the `TRoTR` benchmark by annotating historical text and explicitly modeling change in topical variation over time. This will allow us to track the evolution of a quote like `To be or not to be` where Hamlet originally reflected on the struggles of existence and the fear of the unknown. Over the centuries, the phrase has become deeply embedded in various languages and cultures, often improperly referenced, quoted, and parodied in diverse literary works, contexts, and topics (Bate, 1985).

Limitations. The main limitations of this work pertain to the benchmark, including the data collection and processing:

- *Manual tweet search*: we conducted a manual search of tweets by leveraging the Twitter search bar. This allowed us to sidestep a Text Reuse Detection phase and its validation. However, manually checking the suitability of retrieved tweets is extremely time consuming, thus limiting our ability to collect a large amount of tweets. Moreover, due to the Twitter ranking of matching results, the topic distribution of recontextualizations may be biased.

- *Randomization of the annotation instances*: in generating the pairs of tweets to compare for human judgment, we randomized the order of $\langle t, c_1, c_2 \rangle$ instances. However, we did not randomize the order of the two contexts within a pair. The ordering of c_1 and c_2 in $\langle t, c_1, c_2 \rangle$ was fixed and determined by their IDs. If item order influences annotator judgments, this may have created a bias towards certain orderings.

- *Human judgments*: we discarded some of judgments from human annotators to ensure high-quality of annotation results. This implied a high degree of imbalance in the distribution of TRiC labels for Subtask 1. We addressed and discussed this imbalance in the experimental results (see Section 10.5.2 and Appendix E.1).

As a further limitation, the TRoTR benchmark contains English tweets only with literal text reuse (i.e., explicit quotations). However, the benchmark can be extended to consider multi-language corpora and implicit text reuse.

As this work is the first of its kind to phrase a new problem, recontextualization of text-reuse, create a human-annotated benchmark, and attempt to solve the problem using computational tools, we do not claim our work to be exhaustive.

Ethical considerations. The authors have carefully considered the ethics associated with the TRoTR benchmark. The benchmark data, extracted from Twitter (now X), and annotations have been used while respecting the privacy and confidentiality of both users and annotators. For users, we made an effort to anonymize publicly available tweets' content by removing tweet mentions and users. For human annotators, we explicitly notified them prior to the annotation that some instances of text reuse might encompass discriminatory language against people or communities. We encourage the research community to approach our benchmark with a critical perspective, recognizing the potential ethical implications of working with data from social media platforms.

The annotation campaign was conducted with Native English speakers who were reached through email broadcasts. Compensation details, set in advance, were based on an hourly rate of €12. Each annotator spent a total of 53 hours on the annotation process, resulting in an overall compensation of €636. This fixed compensation was determined according to our time estimation. As per our contract terms, annotators received payment at the conclusion of the annotation campaign.

Chapter 11

Conclusion

> "*So long, and thanks for all the fish*"
>
> Douglas Adams, *The Hitchhiker's Guide to the Galaxy*

In the past five years, the advent of Large Language Models (LLMs) has revolutionized the field of Natural Language Processing (NLP). The capability of LLMs to generate a distinct semantic representation for each occurrence of a target word was considered the most valuable advancement for any text-based researchers (Chernyavskiy et al., 2021). However, although an increasing number of studies have been testing LLMs in *synchronic* scenarios and tasks, few studies have focused on *diachronic* scenarios and semantic change. Thus, this thesis represents a significant contribution to the field of NLP, bridging the gap between the synchronic modeling of word meaning, and the diachronic modeling of their semantic change.

At the beginning of my PhD, word embeddings were considered the preferred tool for modeling word meaning. Thus, this thesis initially placed particular emphasis on encoder-based LLMs (e.g., BERT, mBERT, XLM-R). In later stages, in response to recent advancements in text generation (e.g., GPT, LlaMa), I have expanded my discussion to include and explore the modeling of meaning through generative models. Nonetheless, my discussion is often general and can be applied to different classes of LLMs.

In the introduction of this thesis, we formulated three primary research questions (RQs). We have addressed these RQs throughout the chapters, and we now present a summary of our contributions.

11.1 Summary of contributions

RQ1 *How can lexical semantic change be modeled using LLMs?*

To model lexical semantic change through LLMs, *existing approaches* typically follow a standard recipe. Given a target word and a diachronic corpus spanning two time periods, these approaches i) extract all the usages of the word from the corpus, ii) generate a semantic representation for each word occurrence, iii) optionally aggregate these representations into sense representations, and iv) finally assess the degree of semantic change by applying a distance measure to the word representations from different time periods.

In Chapter 2, we thoroughly reviewed the relevant literature at the beginning of my PhD. In particular, we propose a novel classification framework to categorize existing approaches according to three dimensions of analysis: *meaning representation*, *time awareness*, and *learning modality*. We also discussed performance, open challenges, and main limitations in the current state of the modeling. Among these, we note that modeling lexical semantic change thus far has been approached using several simplifications. Given a word, existing approaches mainly focus on quantifying the change of the *dominant meaning* and are limited to detecting change over *two time periods*. While these simplifications have served as building blocks for studying language change, they prevent from modeling the evolution of *each individual sense of a word over time*, and thus, from answering research questions posed in text-based research fields.

In Chapter 7, we noted that state-of-the-art comparisons are often conducted under varied conditions, which may lead to misleading conclusions. Additionally, we also observed that most of existing approaches have been evaluated on semantic change quantification but not on how they model meaning.

Considering the first issue, we performed a systematic evaluation comparing different LLMs (i.e., BERT, mBERT, XLM-R, XL-LEXEME) and approaches (i.e., APD, PRT, AP+JSD, WiDiD) across multiple languages (i.e., English, Latin, Swedish, German, Spanish, Chinese, Norwegian, Russian) under identical conditions. Our experiments demonstrated that, currently, XL-LEXEME is the most effective LLM for modeling the semantics of word in-context. Our experiments also showed that, in monolingual scenarios, monolingual pre-trained BERT models outperform multilingual pre-trained models such as mBERT and XLM-R. Additionally, we discovered that the standard practice of using word embeddings generated by the last layer of these models is typically not the most effective option for modeling semantic change. Instead, we found that other layers consistently achieve higher performance. Furthermore, we find that approaches that quantify semantic change based on features such as polysemy and dominant word meaning prove to be more powerful than those attempting to model each meaning of a word individually before modeling semantic change.

Considering the second issue, we connected the current modeling of lexical semantic change with other established NLP problems and further evaluated LLMs in tasks such as Word-in-Context and Word Sense Induction. Our experiments demonstrated that while word embeddings perform comparably to human-level in Word-in-Context and Graded Change Detection tasks, they exhibit only medium-low performance in Word Sense Induction.

Since our initial focus was on word embeddings, we investigated alternative semantic representations for word occurrences in Chapters 3, 8, and 9. Specifically, we investigated the use of prompt answers (Chapter 3), lexical replacements and substitutes (Chapter 8), and sense definitions (Chapter 9). Throughout our investigation, we extended our evaluation to generative language models (e.g., GPT-3.5, GPT-4, LLaMA2, LLaMA2-Chat, LLaMA3-Instruct, Flan-T5). Our findings suggest that: (i) while embeddings provide a more scalable solution compared to recent generative models, they often present challenges in terms of interpretability; (ii) prompt-based approaches are inadequate due to limitations in both performance and accessibility; (iii) lexical replacements and substitutes provide interpretability and achieve results that are at least comparable to state-of-the-art performance; (iv) automatically generated sense definitions combined with sentence embeddings represent a promising approach for modeling word meaning, offering improved interpretability.

RQ2 *How can the existing modeling be expanded to handle multiple time periods?*

To expand the existing modeling of lexical semantic change, we challenged the general assumption that approaches proposed for the modeling over two time periods are also suitable over multiple time periods.

In Chapter 4, we presented various strategies to expand the existing modeling towards diachronic word sense induction, aiming to create a diachronic word sense inventory that facilitates both semantic change assessment and interpretations. These strategies include i) clustering word-usage representations from consecutive time intervals, ii) clustering word-usage representations from consecutive time periods, iii) performing one-time clustering of word-usage representations from all time periods, iv) implementing incremental clustering of word-usage representations from consecutive time periods, and v) scaling up clustering with *form*-based approaches. We emphasized that each approach has its advantages and drawbacks, and the choice of modeling should depend on the research questions and available data. However, we believe that modeling lexical semantic change should involve the use of solutions that take the temporal nature of language into account, such as *incremental, evolutionary clustering*.

In this regard, in Chapter 5, we proposed a new algorithm, called A-Posteriori affinity Propagation, that is both *scalable* and *evolutionary*. Through rigorous experimentation, we demonstrate the effectiveness of this algorithm in general clustering settings. We then integrate it into a novel approach for modeling lexical semantic change to facilitate the handling of semantic representations (e.g., word embeddings), and the study of the evolution of each individual word meaning over time. In Chapter 6, we illustrated the application of our approach by considering target words across two Italian datasets containing: i) Italian parliamentary speeches, and ii) Vatican publications, respectively. In Chapter 5 and 6 and 7, we evaluated the use of APP combined to different LMs (i.e., BERT, mBERT, XLM-R, XL-LEXEME) across different languages (e.g., English, Latin, Swedish, German, Spanish, Chinese, Norwegian, Russian, Italian), demonstrating its superiority compared to the current state-of-the-art. Nonetheless, although enhancing the current modeling and state-of-the-art, we relied on several simplification and thus believe there is still ample room for improvement. The incremental modeling of lexical semantic change through LLMs represents a pioneering endeavor in the field of NLP, and as such, we believe it will inspire future research for a more comprehensive modeling of word meaning that incorporates temporal information.

RQ3 *How can the existing modeling be extended to model historical resonance?*

Thus far, historical resonance has been modeled by merely considering the detection of text reuse excerpts (e.g., literary quotations). However, we observe that these approaches do not focus on recontextualization, i.e., how the new context(s) of a reused text differs from its original context(s).

Thus, in Chapter 10, we define historical resonance as *text-reuse re-contextualization* and introduce a novel evaluation framework, called `TRoTR`, to evaluate computational methods and LLMs in capturing the recontextualization of text-reuse. This framework relies on the notion of topic relatedness and consists of two tasks, namely Text Reuse in-Context (TRiC) and Topic Variation Ranking across Corpus (TRaC), which offer two different semantic-change evaluation settings.

To support evaluation, we conducted a human-annotation campaign to collect judgments on topic relatedness over re-contextualizations of biblical passages in tweets, thereby creating an evaluation benchmark with gold standard labels for both TRiC and TRaC tasks. We comprehensively evaluated 36 different SBERT models in different setting (i.e., pre-trained, fine-tuned, and by masking the text reuse instance) to asses their suitability for modeling topic relatedness. Our findings hold true for all these models and indicate that current sequence models are more sensitive to textual similarity rather than topic relatedness. Consequently, different texts containing common substrings are prone to be erroneously considered related in topic due to their shared substrings. Additionally, our results suggests that LLMs trained on Bi-Encoder architectures obtain higher results than LLMs trained on Cross-Encoder architectures.

Bibliography

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif Mohammad. 2023. What Makes Sentences Semantically Related? A Textual Relatedness Dataset and Empirical Study. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 782–796, Dubrovnik, Croatia. Association for Computational Linguistics.

Aly Abdelrazek, Yomna Eid, Eman Gawish, Walaa Medhat, and Ahmed Hassan. 2023. Topic Modeling Algorithms and Applications: A Survey. *Information Systems*, 112:102131.

Mostafa Abdou, Vinit Ravishankar, Artur Kulmizev, and Anders Søgaard. 2022. Word Order Does Matter and Shuffled Language Models Know It. In *Proc. of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6907–6919, Dublin, Ireland. Association for Computational Linguistics.

Oshin Agarwal and Ani Nenkova. 2022. Temporal Effects on Pre-trained Models for Language Processing Tasks. *Transactions of the Association for Computational Linguistics*, 10:904–921.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 Task 10: Multilingual Semantic Textual Similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. SEM 2013 shared task: Semantic Textual Similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Eneko Agirre and Aitor Soroa. 2007. SemEval-2007 Task 02: Evaluating Word Sense Induction and Discrimination Systems. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 7–12, Prague, Czech Republic. Association for Computational Linguistics.

Taichi Aida and Danushka Bollegala. 2023. Swap and Predict – Predicting the Semantic Changes in Words across Corpora by Context Swapping. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7753–7772, Singapore. Association for Computational Linguistics.

Taichi Aida and Danushka Bollegala. 2024. A Semantic Distance Metric Learning approach for Lexical Semantic Change Detection. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 7570–7584, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Anna Aksenova, Ekaterina Gavrishina, Elisei Rykov, and Andrey Kutuzov. 2022. RuDSI: Graph-based Word Sense Induction Dataset for Russian. In *Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing*, pages 77–88, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Domagoj Alagic, Jan Snajder, and Sebastian Pado. 2018. Leveraging Lexical Substitutes for Unsupervised Word Sense Induction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Ian L Alberts, Lorenzo Mercolli, Thomas Pyka, George Prenosil, Kuangyu Shi, Axel Rominger, and Ali Afshar-Oromieh. 2023. Large Language Models (LLM) and ChatGPT: What Will the Impact on Nuclear Medicine Be? *European journal of nuclear medicine and molecular imaging*, 50(6):1549–1552.

Ashjan Alsulaimani and Erwan Moreau. 2023. Improving Diachronic Word Sense Induction with a Non-parametric Bayesian Method. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8908–8925, Toronto, Canada. Association for Computational Linguistics.

Ashjan Alsulaimani, Erwan Moreau, and Carl Vogel. 2020. An Evaluation Method for Diachronic Word Sense Induction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3171–3180, Online. Association for Computational Linguistics.

Fares Antaki, Samir Touma, Daniel Milad, Jonathan El-Khoury, and Renaud Duval. 2023. Evaluating the Performance of ChatGPT in Ophthalmology: An Analysis of Its Successes and Shortcomings. *Ophthalmology Science*, 3(4):100324.

Marianna Apidianaki. 2023. From Word Types to Tokens and Back: A Survey of Approaches to Word Meaning Representation and Interpretation. *Computational Linguistics*, 49(2):465–523.

Nikolay Arefyev, Maksim Fedoseev, Vitaly Protastov, Daniil Homiskiy, Adis Davletov, and Alexander Panchenko. 2021. DeepMistake: Which Senses are Hard to Distinguish for a Word-in-Context Model. In *Proceedings of the Conference on Computational Linguistics and Intellectual Technologies (Dialogue)*, (online). RSUH.

Nikolay Arefyev and Vasily Zhikov. 2020. BOS at SemEval-2020 Task 1: Word Sense Induction via Lexical Substitution for Lexical Semantic Change Detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 171–179, Barcelona (online). International Committee for Computational Linguistics.

Carlos Santos Armendariz, Matthew Purver, Senja Pollak, Nikola Ljubešić, Matej Ulčar, Ivan Vulić, and Mohammad Taher Pilehvar. 2020a. SemEval-2020 Task 3: Graded Word Similarity in Context. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 36–49, Barcelona (online). International Committee for Computational Linguistics.

Carlos Santos Armendariz, Matthew Purver, Matej Ulčar, Senja Pollak, Nikola Ljubešić, and Mark Granroth-Wilding. 2020b. CoSimLex: A Resource for Evaluating Graded Word Similarity in Context. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5878–5886, Marseille, France. European Language Resources Association.

Natalia M. Arzeno and Haris Vikalo. 2017. Evolutionary Affinity Propagation. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2681–2685.

Natalia M. Arzeno and Haris Vikalo. 2021. Evolutionary Clustering via Message Passing. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 33(6):2452–2466.

Tal August, Katharina Reinecke, and Noah A. Smith. 2022. Generating Scientific Definitions with Controllable Complexity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8298–8317, Dublin, Ireland. Association for Computational Linguistics.

Anthony Baez and Horacio Saggion. 2023. LSLlama: Fine-Tuned LLaMA for Lexical Simplification. In *Proceedings of the Second Workshop on Text Simplification, Accessibility and Readability*, pages 102–108, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Bing Bai, Jian Liang, Guanhua Zhang, Hao Li, Kun Bai, and Fei Wang. 2021. Why Attentions May Not Be Interpretable? In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 25–34, New York, NY, USA. Association for Computing Machinery.

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondrej Dusek. 2024. Leak, Cheat, Repeat: Data Contamination and Evaluation Malpractices in Closed-Source LLMs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 67–93, St. Julian's, Malta. Association for Computational Linguistics.

David Bamman and Patrick J. Burns. 2020. Latin BERT: A Contextual Language Model for Classical Philology.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718, Nusa Dua, Bali. Association for Computational Linguistics.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. 2004. Correlation clustering. *Machine learning*, 56:89–113.

Pierpaolo Basile, Annalina Caputo, Tommaso Caselli, Pierluigi Cassotti, and Rossella Varvara. 2020. DIACR-Ita@ EVALITA2020: Overview of the EVALITA2020 DiachronicLexical Semantics (DIACR-Ita) Task. In *Proceedings of the Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA)*, Online. CEUR-WS.

Jonathan Bate. 1985. Parodies of Shakespeare. *Journal of Popular Culture*, 19(1):75.

Nina Bauwelinck and Els Lefever. 2020. Annotating Topics, Stance, Argumentativeness and Claims in Dutch Social Media Comments: A Pilot Study. In *Proceedings of the 7th Workshop on Argument Mining*, pages 8–18, Online. Association for Computational Linguistics.

Diego Bear and Paul Cook. 2021. Cross-Lingual Wolastoqey-English Definition Modelling. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 138–146, Held Online. INCOMA Ltd.

Christin Beck. 2020. DiaSense at SemEval-2020 Task 1: Modeling Sense Change via Pre-trained BERT Embeddings. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 50–58, Barcelona (online). International Committee for Computational Linguistics.

Jürgen Beringer and Eyke Hüllermeier. 2006. Online Clustering of Parallel Data Streams. *Data & Knowledge Engineering (DKE)*, 58(2):180–204.

Michele Bevilacqua, Marco Maru, and Roberto Navigli. 2020. Generationary or "How We Went beyond Word Sense Inventories and Learned to Gloss". In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7207–7221, Online. Association for Computational Linguistics.

Andreas Blank. 1997. *Prinzipien des lexikalischen Bedeutungswandels am Beispiel der romanischen Sprachen*. Max Niemeyer Verlag, Berlin, Boston.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3(null):993–1022.

Terra Blevins and Luke Zettlemoyer. 2020. Moving Down the Long Tail of Word Sense Disambiguation with Gloss Informed Bi-encoders. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1006–1017, Online. Association for Computational Linguistics.

Leonard Bloomfield. 1933. *Language*. Holt, Rinehart and Winston, New York.

John W. Du Bois. 2014. Towards a Dialogic Syntax. *Cognitive Linguistics*, 25(3):359–410.

Brian Bonafilia, Bastiaan Bruinsma, Denitsa Saynova, and Moa Johansson. 2023. Sudden Semantic Shifts in Swedish NATO Discourse. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 184–193, Toronto, Canada. Association for Computational Linguistics.

Francis Bond and Graham Matthews. 2018. Toward An Epic Epigraph Graph. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Nicolas Bonneel, Michiel van de Panne, Sylvain Paris, and Wolfgang Heidrich. 2011. Displacement Interpolation using Lagrangian Mass Transport. *ACM Trans. Graph.*, 30(6):1–12.

Emanuela Boros, Maud Ehrmann, Matteo Romanello, Sven Najem-Meyer, and Frédéric Kaplan. 2024. Post-Correction of Historical Text Transcripts with Large Language Models: An Exploratory Study. In *Proceedings of the 8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2024)*, pages 133–159, St. Julians, Malta. Association for Computational Linguistics.

Michel Bréal. 1904. *Essai de Sémantique (Science des Significations)*. Hachette.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Marco Büchler, Philip R. Burns, Martin Müller, Emily Franzini, and Greta Franzini. 2014. *Towards a Historical Text Re-use Detection*, pages 221–238. Springer International Publishing, Cham.

Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. 2021. Isotropy in the Contextual Embedding Space: Clusters and Manifolds. In *International Conference on Learning Representations*.

Lyle Campbell. 2020. *Historical Linguistics*. Edinburgh University Press, Edinburgh.

Dallas Card. 2023. Substitution-based Semantic Change Detection using Contextual Embeddings. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 590–602, Toronto, Canada. Association for Computational Linguistics.

Pierluigi Cassotti, Lucia Siciliani, Marco DeGemmis, Giovanni Semeraro, and Pierpaolo Basile. 2023a. XL-LEXEME: WiC Pretrained Model for Cross-Lingual LEXical sEMantic changE. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1577–1585, Toronto, Canada. Association for Computational Linguistics.

Pierluigi Cassotti, Lucia Siciliani, Lucia Passaro, Maristella Gatto, and Pierpaolo Basile. 2023b. WiC-ITA at EVALITA2023: Overview of the EVALITA2023 Word-in-Context for ITAlian Task. In *Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA 2023)*, Parma, Italy. CEUR.org.

Silvana Castano, Alfio Ferrara, Stefano Montanelli, and Francesco Periti. 2024. Incremental Affinity Propagation based on Cluster Consolidation and Stratification.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Deepayan Chakrabarti, Ravi Kumar, and Andrew Tomkins. 2006. Evolutionary Clustering. In *Proceedings of the 12th ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 554–560, Philadelphia, PA, USA. Association for Computing Machinery.

Ming-Wei Chang, Lev Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of Semantic Representation: Dataless Classification. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, page 830–835, Chicago, Illinois. AAAI Press.

Ting-Yun Chang and Yun-Nung Chen. 2019. What Does This Word Mean? Explaining Contextualized Embeddings with Natural Language Definition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6064–6070, Hong Kong, China. Association for Computational Linguistics.

Chi Cheang, Hou Chan, Derek Wong, Xuebo Liu, Zhaocong Li, Yanming Sun, Shudong Liu, and Lidia Chao. 2023. Can LMs Generalize to Future Data? An Empirical Analysis on Text Summarization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16205–16217, Singapore. Association for Computational Linguistics.

Jing Chen, Emmanuele Chersoni, and Chu-ren Huang. 2022a. Lexicon of Changes: Towards the Evaluation of Diachronic Semantic Shift in Chinese. In *Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change*, pages 113–118, Dublin, Ireland. Association for Computational Linguistics.

Jing Chen, Emmanuele Chersoni, Dominik Schlechtweg, Jelena Prokic, and Chu-Ren Huang. 2023a. ChiWUG: A Graph-based Evaluation Dataset for Chinese Lexical Semantic Change Detection. In *Proceedings of the 4th Workshop on Computational Approaches to Historical Language Change*, pages 93–99, Singapore. Association for Computational Linguistics.

Jing Chen, Emmanuele Chersoni, Dominik Schlechtweg, Jelena Prokic, and Chu-Ren Huang. 2023b. ChiWUG: Diachronic Word Usage Graphs for Chinese.

Ze Chen, Kangxu Wang, Zijian Cai, Jiewen Zheng, Jiarong He, Max Gao, and Jason Zhang. 2022b. Using Deep Mixture-of-Experts to Detect Word Meaning Shift for TempoWiC. In *Proceedings of the First Workshop on Ever Evolving NLP (EvoNLP)*, pages 7–11, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Pauline Hope Cheong. 2014. Tweet the Message? Religious Authority and Social Media Innovation. *Journal of Religion, Media and Digital Culture*, 3(3):1 – 19.

Anton Chernyavskiy, Dmitry Ilvovsky, and Preslav Nakov. 2021. Transformers: "The End of History" for Natural Language Processing? In *Machine Learning and Knowledge Discovery in Databases. Research Track*, pages 677–693, Cham. Springer International Publishing.

Cheng-Han Chiang and Hung-yi Lee. 2023. Are Synonym Substitution Attacks Really Synonym Substitution Attacks? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1853–1878, Toronto, Canada. Association for Computational Linguistics.

Ting-Rui Chiang and Dani Yogatama. 2023. The Distributional Hypothesis Does Not Fully Explain the Benefits of Masked Language Model Pretraining. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10305–10321, Singapore. Association for Computational Linguistics.

Stanford Chiu, Ibrahim Uysal, and W. Bruce Croft. 2010. Evaluating Text Reuse Discovery on the Web. In *Proceedings of the Third Symposium on Information Interaction in Context*, IIiX '10, page 299–304, New Brunswick, New Jersey, USA. Association for Computing Machinery.

Hyunjin Choi, Judong Kim, Seongho Joe, Seungjai Min, and Youngjune Gwon. 2021. Analyzing Zero-shot Cross-lingual Transfer in Supervised NLP Tasks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 9608–9613.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2024. Scaling Instruction-Finetuned Language Models. *Journal of Machine Learning Research*, 25(70):1–53.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What Does BERT Look at? An Analysis of BERT's Attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Paul Clough, Robert Gaizauskas, Scott S.L. Piao, and Yorick Wilks. 2002. Measuring Text Reuse. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 152–159, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

John H. Connolly. 2014. Recontextualisation, Resemiotisation and Their Analysis in Terms of an FDG-based Framework. *Pragmatics*, 24(2):377–397.

Paul Cook, Jey Han Lau, Diana McCarthy, and Timothy Baldwin. 2014. Novel Word-sense Identification. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1624–1635, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

D. Alan Cruse. 2000. *Aspects of the Micro-Structure of Word Meanings*, pages 30–51. Oxford University Press.

Amaru Cuba Gyllensten, Evangelia Gogoulou, Ariel Ekgren, and Magnus Sahlgren. 2020. SenseCluster at SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 112–118, Barcelona (online). International Committee for Computational Linguistics.

Miriam Cuscito, Alfio Ferrara, and Martin Ruskov. 2024. How BERT Speaks Shakespearean English? Evaluating Historical Bias in Contextual Language Models.

Arséne Darmesteter. 1893. *La Vie des Mots Étudiée Dans Leurs Significations*. C. Delagrave.

Marco Del Tredici, Raquel Fernández, and Gemma Boleda. 2019. Short-Term Meaning Shift: A Distributional Exploration. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2069–2075, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhijie Deng, Yucen Luo, and Jun Zhu. 2019. Cluster Alignment With a Teacher for Unsupervised Domain Adaptation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9943–9952.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Finetuning of Quantized LLMs.

David DeVault and Matthew Stone. 2004. Interpreting Vague Utterances in Context. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 1247–1253, Geneva, Switzerland. COLING.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Wai Chee Dimock. 1997. A Theory of Resonance. *PMLA*, 112(5):1060–1071.

George Doddington. 2002. Automatic Evaluation of Machine Translation Quality Using n-gram Co-occurrence Statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, page 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Haim Dubossarsky, Simon Hengchen, Nina Tahmasebi, and Dominik Schlechtweg. 2019. Time-Out: Temporal Referencing for Robust Modeling of Lexical Semantic Change. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 457–470, Florence, Italy. Association for Computational Linguistics.

Haim Dubossarsky, Yulia Tsvetkov, Chris Dyer, and Eitan Grossman. 2015. A Bottom Up Approach to Category Mapping and Meaning Change. In *Proceedings of the NetWordS Final Conference*, pages 66–70, Pisa, Italy. CEUR-WS.

Haim Dubossarsky, Daphna Weinshall, and Eitan Grossman. 2017. Outta Control: Laws of Semantic Change and Inherent Biases in Word Representation Models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1136–1145, Copenhagen, Denmark. Association for Computational Linguistics.

Martin Emms and Arun Kumar Jayapal. 2016. Dynamic Generative model for Diachronic Sense Emergence Detection. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1362–1373, Osaka, Japan. The COLING 2016 Organizing Committee.

Volkmar Engerer. 2017. Exploring Interdisciplinary Relationships between Linguistics and Information Retrieval from the 1960s to Today. *Journal of the Association for Information Science and Technology*, 68(3):660–680.

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. 2013. Measuring Word Meaning in Context. *Computational Linguistics*, 39(3):511–554.

Kawin Ethayarajh. 2019. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Alessio Ferrari, Beatrice Donati, and Stefania Gnesi. 2017. Detecting Domain-Specific Ambiguities: An NLP Approach Based on Wikipedia Crawling and Word Embeddings. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, pages 393–399.

Tony Finch. 2009. Incremental Calculation of Weighted Mean and Variance. *University of Cambridge*, 4(11-5):41–42.

John Rupert Firth. 1957. A Synopsis of Linguistic Theory. *Studies in linguistic analysis*.

Lauren Fonteyn, F Karsdorp, B McGillivray, A Nerghens, and M Wevers. 2020. What About Grammar? Using BERT Embeddings to Explore Functional-Semantic Shifts of Semi-Lexical and Grammatical Constructions. In *Proceedings of the Workshop on Computational Humanities Research (CHR)*, pages 257–268, Amsterdam, the Netherlands. CEUR-WS.

Clémentine Fourrier and Syrielle Montariol. 2022. Caveats of Measuring Semantic Change of Cognates and Borrowings using Multilingual Word Embeddings. In *Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change*, pages 97–112, Dublin, Ireland. Association for Computational Linguistics.

Greta Franzini, Marco Passarotti, Maria Moritz, and Marco Büchler. 2018. Using and Evaluating TRACER for an Index Fontium Computatus of the Summa contra Gentiles of Thomas Aquinas. In *Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Torino, Italy, December 10-12, 2018*, volume 2253 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Brendan J Frey and Delbert Dueck. 2007. Clustering by Passing Messages Between Data Points. *science*, 315(5814):972–976.

Artyom Gadetsky, Ilya Yakubovskiy, and Dmitry Vetrov. 2018. Conditional Generators of Words Definitions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 266–271, Melbourne, Australia. Association for Computational Linguistics.

Noah Gardner, Hafiz Khan, and Chih-Cheng Hung. 2022. Definition Modeling: Literature Review and Dataset Analysis. *Applied Computing and Intelligence*, 2(1):83–98.

Aina Garí Soler and Marianna Apidianaki. 2021. Let's Play Mono-Poly: BERT Can Reveal Words' Polysemy Level and Partitionability into Senses. *Transactions of the Association for Computational Linguistics*, 9:825–844.

Aina Garí Soler, Matthieu Labeau, and Chloé Clavel. 2022. One Word, Two Sides: Traces of Stance in Contextualized Word Representations. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3950–3959, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Dirk Geeraerts. 2020. Semantic Change: "What The Smurf?". *The Wiley Blackwell Companion to Semantics*, pages 1–24.

A. Shaji George and A. S. Hovan George. 2023. A Review of ChatGPT AI's Impact on Several Business Sectors. *Partners Universal International Innovation Journal*, 1(1):9–23.

Sayan Ghosh and Shashank Srivastava. 2022. ePiC: Employing Proverbs in Context as a Benchmark for Abstract Language Understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3989–4004, Dublin, Ireland. Association for Computational Linguistics.

Mario Giulianelli, Marco Del Tredici, and Raquel Fernández. 2020. Analysing Lexical Semantic Change with Contextualised Word Representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3960–3973, Online. Association for Computational Linguistics.

Mario Giulianelli, Andrey Kutuzov, and Lidia Pivovarova. 2022. Do Not Fire the Linguist: Grammatical Profiles Help Language Models Detect Semantic Change. In *Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change*, pages 54–67, Dublin, Ireland. Association for Computational Linguistics.

Mario Giulianelli, Iris Luden, Raquel Fernandez, and Andrey Kutuzov. 2023. Interpretable Word Sense Representations via Definition Generation: The Case of Semantic Change Analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3130–3148, Toronto, Canada. Association for Computational Linguistics.

Mihir Godbole, Parth Dandavate, and Aditya Kane. 2022. Temporal Word Meaning Disambiguation using TimeLMs. In *Proceedings of the The First Workshop on Ever Evolving NLP (EvoNLP)*, pages 55–60, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Hila Gonen, Ganesh Jawahar, Djamé Seddah, and Yoav Goldberg. 2020. Simple, Interpretable and Stable Method for Detecting Words with Usage Change across Corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 538–555, Online. Association for Computational Linguistics.

Roksana Goworek and Haim Dubossarsky. 2024. Toward Sentiment Aware Semantic Change Analysis. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 350–357, St. Julian's, Malta. Association for Computational Linguistics.

Chris Greenough. 2021. From Biblical Text to Twitter: Teaching Biblical Studies in the Zeitgeist of# MeToo. *Journal of Feminist Studies in Religion*, 37(1):133–135.

Stefan Grondelaers, Dirk Speelman, and Dirk Geeraerts. 2010. Lexical Variation and Change. In *The Oxford Handbook of Cognitive Linguistics*. Oxford University Press.

Maarten Grootendorst. 2022. BERTopic: Neural Topic Modeling with a Class-based TF-IDF Procedure.

Yue Guan, Jingwen Leng, Chao Li, Quan Chen, and Minyi Guo. 2020. How Far Does BERT Look At: Distance-based Clustering and Analysis of BERT's Attention. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3853–3860, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Kristina Gulordava and Marco Baroni. 2011. A Distributional Similarity Approach to the Detection of Semantic Change in the Google Books Ngram Corpus. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 67–71, Edinburgh, UK. Association for Computational Linguistics.

Janosch Haber and Massimo Poesio. 2021. Patterns of Polysemy and Homonymy in Contextualised Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2663–2676, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Muhammad Usman Hadi, Qasem al Tashi, Rizwan Qureshi, Abbas Shah, Amgad Muneer, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, and Mubarak Shah. 2023. Large Language Models: A Comprehensive Survey of its Applications, Challenges, Limitations, and Future Prospects.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, pages 1735–1742. IEEE Computer Society.

Matthias Hagen and Benno Stein. 2011. Candidate Document Retrieval for Web-Scale Text Reuse Detection. In *String Processing and Information Retrieval*, pages 356–367, Berlin, Heidelberg. Springer Berlin Heidelberg.

Helena Halmari. 2011. Political Correctness, Euphemism, and Language Change: The Case of 'People First'. *Journal of Pragmatics*, 43(3):828–840. The Language of Space and Time.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501, Berlin, Germany. Association for Computational Linguistics.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey.

Michael Hanna and David Mareček. 2021. Analyzing BERT's Knowledge of Hypernymy via Prompting. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 275–282, Punta Cana, Dominican Republic. Association for Computational Linguistics.

David Harbecke, Yuxuan Chen, Leonhard Hennig, and Christoph Alt. 2022. Why only Micro-F1? Class Weighting of Measures for Relation Classification. In *Proceedings of NLP Power! The First Workshop on Efficient Benchmarking in NLP*, pages 32–41, Dublin, Ireland. Association for Computational Linguistics.

Zellig S. Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Simon Hengchen, Nina Tahmasebi, Dominik Schlechtweg, and Haim Dubossarsky. 2021. *Challenges for Computational Lexical Semantic Change*, pages 341–372. Language Science Press, Berlin.

Tomáš Hercig and Pavel Kral. 2021. Evaluation Datasets for Cross-lingual Semantic Textual Similarity. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 524–529, Held Online. INCOMA Ltd.

Niclas Hertzberg, Robin Cooper, Elina Lindgren, Björn Rönnerstrand, Gregor Rettenegger, Ellen Breitholtz, and Asad Sayeed. 2022. Distributional properties of political dogwhistle representations in Swedish BERT. In *Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*, pages 170–175, Seattle, Washington (Hybrid). Association for Computational Linguistics (ACL).

Jack Hessel and Alexandra Schofield. 2021. How effective is BERT without word ordering? Implications for language understanding and data privacy. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 204–211, Online. Association for Computational Linguistics.

John Hewitt and Percy Liang. 2019. Designing and Interpreting Probes with Control Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

Hans Henrich Hock and Brian D. Joseph. 2019. *Language History, Language Change, and Language Relationship*. De Gruyter Mouton, Berlin, Boston.

Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2021. Dynamic Contextualized Word Embeddings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6970–6984, Online. Association for Computational Linguistics.

Peter Uwe Hohendahl and Marc Silberman. 1977. Introduction to Reception Aesthetics. *New German Critique*, 10:29–63.

Franziska Horn. 2021. Exploring Word Usage Change with Continuously Evolving Embeddings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 290–297, Online. Association for Computational Linguistics.

Eduard Hovy and Chin-Yew Lin. 1998. Automated Text Summarization and the Summarist System. In *TIPSTER TEXT PROGRAM PHASE III: Proceedings of a Workshop held at Baltimore, Maryland, October 13-15, 1998*, pages 197–214, Baltimore, Maryland, USA. Association for Computational Linguistics.

Eduardo Raul Hruschka, Ricardo J. G. B. Campello, Alex A. Freitas, and André C. Ponce Leon F. de Carvalho. 2009. A Survey of Evolutionary Algorithms for Clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 39(2):133–155.

Beizhe Hu, Qiang Sheng, Juan Cao, Yongchun Zhu, Danding Wang, Zhengjia Wang, and Zhiwei Jin. 2023. Learn over Past, Evolve for Future: Forecasting Temporal Trends for Fake News Detection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 116–125, Toronto, Canada. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models.

Renfen Hu, Shen Li, and Shichen Liang. 2019. Diachronic Sense Modeling with Deep Contextualized Word Embeddings: An Ecological View. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3899–3908, Florence, Italy. Association for Computational Linguistics.

Han Huang, Tomoyuki Kajiwara, and Yuki Arase. 2021. Definition Modelling for Appropriate Specificity. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2499–2509, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiaolei Huang and Michael J. Paul. 2019. Neural Temporality Adaptation for Document Classification: Diachronic Word Embeddings and Domain Adaptation Models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4113–4123, Florence, Italy. Association for Computational Linguistics.

Lawrence Hubert and Phipps Arabie. 1985. Comparing Partitions. *Journal of classification*, 2:193–218.

Mathew Huerta-Enochian. 2024. Instruction Fine-Tuning: Does Prompt Loss Matter?

Shotaro Ishihara and Hono Shirai. 2022. Nikkei at SemEval-2022 Task 8: Exploring BERT-based Bi-Encoder Approach for Pairwise Multilingual News Article Similarity. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1208–1214, Seattle, United States. Association for Computational Linguistics.

Shotaro Ishihara, Hiromu Takahashi, and Hono Shirai. 2022. Semantic Shift Stability: Efficient Way to Detect Performance Degradation of Word Embeddings and Pre-trained Language Models. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 205–216, Online only. Association for Computational Linguistics.

Shonosuke Ishiwatari, Hiroaki Hayashi, Naoki Yoshinaga, Graham Neubig, Shoetsu Sato, Masashi Toyoda, and Masaru Kitsuregawa. 2019. Learning to Describe Unknown Phrases with Local and Global Contexts. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3467–3476, Minneapolis, Minnesota. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What Does BERT Learn about the Structure of Language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Christopher Jenkins, Filip Miletic, and Sabine Schulte im Walde. 2023. To Split or Not to Split: Composing Compounds in Contextual Vector Spaces. In *Proceedings of the 2023 Conference on Empirical*

Methods in Natural Language Processing, pages 16131–16136, Singapore. Association for Computational Linguistics.

Ming Jiang, Yuerong Hu, Glen Worthey, Ryan C Dubnicek, Ted Underwood, and J Stephen Downie. 2021. Impact of OCR Quality on BERT Embeddings in the Domain Classification of Book Excerpts. In *Proceedings of the Conference on Computational Humanities Research 2021*, Amsterdam, the Netherlands.

Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, and Zhaopeng Tu. 2023. Is ChatGPT A Good Translator? Yes With GPT-4 As The Engine.

Xianlin Jin and Patric R. Spence. 2021. Understanding crisis communication on social media with CERC: topic model analysis of tweets about Hurricane Maria. *Journal of Risk Research*, 24(10):1266–1287.

John S. Justeson and Slava M. Katz. 1991. Co-occurrences of Antonymous Adjectives and Their Contexts. *Computational Linguistics*, 17(1):1–20.

Alexander Kalinowski and Yuan An. 2021. Exploring Sentence Embedding Structures for Semantic Relation Extraction. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7.

Masahiro Kaneko and Danushka Bollegala. 2021. Debiasing Pre-trained Contextualised Embeddings. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1256–1266, Online. Association for Computational Linguistics.

Vani Kanjirangat, Sandra Mitrovic, Alessandro Antonucci, and Fabio Rinaldi. 2020. SST-BERT at SemEval-2020 Task 1: Semantic Shift Tracing by Clustering in BERT-based Embedding Spaces. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 214–221, Barcelona (online). International Committee for Computational Linguistics.

Nikitas Karanikolas, Eirini Manga, Nikoletta Samaridi, Eleni Tousidou, and Michael Vassilakopoulos. 2024. Large Language Models versus Natural Language Understanding and Generation. In *Proceedings of the 27th Pan-Hellenic Conference on Progress in Computing and Informatics*, PCI '23, page 278–290, Lamia, Greece. Association for Computing Machinery.

Andres Karjus. 2023. Machine-assisted Mixed Methods: Augmenting Humanities and Social Sciences with Artificial Intelligence.

Anna Karnysheva and Pia Schwarz. 2020. TUE at SemEval-2020 Task 1: Detecting Semantic Change by Clustering Contextual Word Embeddings. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 232–238, Barcelona (online). International Committee for Computational Linguistics.

Kseniia Kashleva, Alexander Shein, Elizaveta Tukhtina, and Svetlana Vydrina. 2022. HSE at LSCDiscovery in Spanish: Clustering and Profiling for Lexical Semantic Change Discovery. In *Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change*, pages 193–197, Dublin, Ireland. Association for Computational Linguistics.

Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. 2024. A Survey of Reinforcement Learning from Human Feedback.

Margarita Kay. 1979. Lexemic Change and Semantic Shift in Disease Names. *Culture, medicine and psychiatry*, 3(1):73–94.

Raef Kazi, Alessandra Amato, Shenghui Wang, and Doina Bucur. 2022. Visualisation Methods for Diachronic Semantic Shift. In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 89–94, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Daphna Keidar, Andreas Opedal, Zhijing Jin, and Mrinmaya Sachan. 2022. Slangvolution: A Causal Analysis of Semantic Change and Frequency Dynamics in Slang. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1422–1442, Dublin, Ireland. Association for Computational Linguistics.

Olga Kellert and Md Mahmud Uz Zaman. 2022. Using Neural Topic Models to Track Context Shifts of Words: a Case Study of COVID-related Terms Before and After the Lockdown in April 2020. In *Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change*, pages 131–139, Dublin, Ireland. Association for Computational Linguistics.

Mohammad Khalil and Erkan Er. 2023. Will ChatGPT Get You Caught? Rethinking of Plagiarism Detection. In *Learning and Collaboration Technologies: 10th International Conference, LCT 2023, Held as Part of the 25th HCI International Conference, HCII 2023, Copenhagen, Denmark, July 23–28, 2023, Proceedings, Part I*, page 475–487, Copenhagen, Denmark. Springer-Verlag.

Anton S. Khritankov, Pavel V. Botov, Nikolay S. Surovenko, Sergey V. Tsarkov, Dmitriy V. Viuchnov, and Yuri V. Chekhovich. 2015. Discovering Text Reuse in Large Collections of Documents: A Study of Theses in History Sciences. In *2015 Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference (AINL-ISMW FRUCT)*, pages 26–32.

Keivan Kianmehr, Mohammed Alshalalfa, and Reda Alhajj. 2010. Fuzzy clustering-based discretization for gene expression classification. *Knowledge and Information Systems*, 24:441–465.

Adam Kilgarriff. 1997. I Don't Believe in Word Senses. *Computers and the Humanities*, 31(2):91–113.

Munui Kim, Injun Baek, and Min Song. 2018. Topic Diffusion Analysis of a Weighted Citation Network in Biomedical Literature. *Journal of the Association for Information Science and Technology*, 69(2):329–342.

Kazuma Kobayashi, Taichi Aida, and Mamoru Komachi. 2021. Analyzing Semantic Changes in Japanese Words Using BERT. In *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*, pages 270–280, Shanghai, China. Association for Computational Lingustics.

Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, Anna Kocoń, Bartłomiej Koptyra, Wiktoria Mieleszczenko-Kowszewicz, Piotr Miłkowski, Marcin Oleksy, Maciej Piasecki, Łukasz Radliński, Konrad Wojtasik, Stanisław Woźniak, and Przemysław Kazienko. 2023. ChatGPT: Jack of All Trades, Master of None. *Information Fusion*, 99:101861.

Cunliang Kong, Yun Chen, Hengyuan Zhang, Liner Yang, and Erhong Yang. 2022. Multitasking Framework for Unsupervised Simple Definition Generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5934–5943, Dublin, Ireland. Association for Computational Linguistics.

Miloslav Konopík, Ondřej Pražák, and David Steinberger. 2017. Czech Dataset for Semantic Similarity and Relatedness. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 401–406, Varna, Bulgaria. INCOMA Ltd.

Matej Kosec, Sheng Fu, and Mario Michael Krell. 2021. Packing: Towards 2x NLP BERT Acceleration.

Klaus Krippendorff. 2019. *Content Analysis*. SAGE Publications, Inc.

Artem Kudisov and Nikolay Arefyev. 2022. BOS at LSCDiscovery: Lexical Substitution for Interpretable Lexical Semantic Change Detection. In *Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change*, pages 165–172, Dublin, Ireland. Association for Computational Linguistics.

Parag A. Kulkarni and Preeti Mulay. 2013. Evolve Systems using Incremental Clustering Approach. *Evolving Systems*, 4(2):71–85.

Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2015. Statistically Significant Detection of Linguistic Change. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, page 625–635, Florence, Italy. International World Wide Web Conferences Steering Committee.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language. In *Proceedings of the Conference on Computational Linguistics and Intellectual Technologies (Dialogue)*, Moscow, Russia. RSUH.

Andrey Kutuzov. 2020. *Distributional Word Embeddings in Modeling Diachronic Semantic Change*. Ph.D. thesis, University of Oslo.

Andrey Kutuzov and Mario Giulianelli. 2020. UiO-UvA at SemEval-2020 Task 1: Contextualised Embeddings for Lexical Semantic Change Detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 126–134, Barcelona (online). International Committee for Computational Linguistics.

Andrey Kutuzov, Elizaveta Kuzmenko, and Lidia Pivovarova. 2017. Clustering of Russian Adjective-Noun Constructions using Word Embeddings. In *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing*, pages 3–13, Valencia, Spain. Association for Computational Linguistics.

Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. 2018. Diachronic Word Embeddings and Semantic Shifts: a Survey. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1384–1397, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Andrey Kutuzov and Lidia Pivovarova. 2021a. RuShiftEval.

Andrey Kutuzov and Lidia Pivovarova. 2021b. RuShiftEval: A Shared Task on Semantic Shift Detection for Russian. In *Proceedings of the Conference on Computational Linguistics and Intellectual Technologies (Dialogue)*, 20, (online). RSUH.

Andrey Kutuzov and Lidia Pivovarova. 2021c. Three-part Diachronic Semantic Change Dataset for Russian. In *Proceedings of the 2nd International Workshop on Computational Approaches to Historical Language Change 2021*, pages 7–13, Online. Association for Computational Linguistics.

Andrey Kutuzov, Lidia Pivovarova, and Mario Giulianelli. 2021a. Grammatical Profiling for Semantic Change Detection. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 423–434, Online. Association for Computational Linguistics.

Andrey Kutuzov, Samia Touileb, Petter Mæhlum, Tita Enstad, and Alexandra Wittemann. 2022a. NorDiaChange: Diachronic Semantic Change Dataset for Norwegian. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2563–2572, Marseille, France. European Language Resources Association.

Andrey Kutuzov, Samia Touileb, Petter Mæhlum, Tita Ranveig Enstad, and Alexandra Wittemann. 2021b. NorDiaChange.

Andrey Kutuzov, Erik Velldal, and Lilja Øvrelid. 2022b. Contextualized Embeddings for Semantic Change Detection: Lessons Learned. *Northern European Journal of Language Technology, Volume 8*.

Caterina Lacerra, Tommaso Pasini, Rocco Tripodi, and Roberto Navigli. 2021. ALaSca: an Automated approach for Large-Scale Lexical Substitution. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3836–3842. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Viet Lai, Nghia Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Nguyen. 2023. ChatGPT Beyond English: Towards a Comprehensive Evaluation of Large Language Models in Multilingual Learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13171–13189, Singapore. Association for Computational Linguistics.

Severin Laicher, Gioia Baldissin, Enrique Castañeda, Dominik Schlechtweg, and Sabine Schulte. 2020. CL-IMS @ DIACR-Ita: Volente o Nolente: BERT does not Outperform SGNS on Semantic Change Detection. In *Proc. of the Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA)*, pages 438–443, Marrakech, Morocco. CEUR-WS.

Severin Laicher, Sinan Kurtyigit, Dominik Schlechtweg, Jonas Kuhn, and Sabine Schulte im Walde. 2021. Explaining and Improving BERT Performance on Lexical Semantic Change Detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 192–202, Online. Association for Computational Linguistics.

Brenden M Lake and Gregory L Murphy. 2023. Word Meaning in Minds and Machines. *Psychological Review*, 130(2):401–431.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Huang. 2023. A Systematic Study and Comprehensive Evaluation of ChatGPT on Benchmark Datasets. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 431–469, Toronto, Canada. Association for Computational Linguistics.

Jey Han Lau, Paul Cook, Diana McCarthy, David Newman, and Timothy Baldwin. 2012. Word Sense Induction for Novel Sense Detection. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 591–601, Avignon, France. Association for Computational Linguistics.

Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China. PMLR.

Hanbit Lee, Yeonchan Ahn, Haejun Lee, Seungdo Ha, and Sang-goo Lee. 2016. Quote Recommendation in Dialogue Using Deep Neural Network. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, page 957–960, Pisa, Italy. Association for Computing Machinery.

Hyun Seung Lee, Seungtaek Choi, Yunsung Lee, Hyeongdon Moon, Shinhyeok Oh, Myeongho Jeong, Hyojun Go, and Christian Wallraven. 2023. Cross Encoding as Augmentation: Towards Effective Educational Text Classification. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2184–2195, Toronto, Canada. Association for Computational Linguistics.

Piroska Lendvai and Claudia Wick. 2022. Finetuning Latin BERT for Word Sense Disambiguation on the Thesaurus Linguae Latinae. In *Proceedings of the Workshop on Cognitive Aspects of the Lexicon*, pages 37–41, Taipei, Taiwan. Association for Computational Linguistics.

Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. SenseBERT: Driving Some Sense into BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4656–4667, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Cheng Li, Jindong Wang, Yixuan Zhang, Kaijie Zhu, Wenxin Hou, Jianxun Lian, Fang Luo, Qiang Yang, and Xing Xie. 2023. Large Language Models Understand and Can be Enhanced by Emotional Stimuli.

Meng Liang and Yao Shi. 2023. Named Entity Recognition Method Based on BERT-whitening and Dynamic Fusion Model. In *2023 5th International Conference on Natural Language Processing (ICNLP)*, pages 191–197.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Ruixi Lin and Hwee Tou Ng. 2022. Does BERT Know that the IS-A Relation Is Transitive? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 94–99, Dublin, Ireland. Association for Computational Linguistics.

Per Linell. 1998. *Approaching Dialogue: Talk, Interaction and Contexts in Dialogical Perspectives*. John Benjamins.

Zhidong Ling, Taichi Aida, Teruaki Oka, and Mamoru Komachi. 2023. Construction of Evaluation Dataset for Japanese Lexical Semantic Change Detection. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 125–136, Hong Kong, China. Association for Computational Linguistics.

Qianchu Liu, Edoardo Maria Ponti, Diana McCarthy, Ivan Vulić, and Anna Korhonen. 2021a. AM2iCo: Evaluating Word Meaning in Context across Low-Resource Languages with Adversarial Examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7151–7162, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yang Liu, Alan Medlar, and Dorota Glowacka. 2021b. Statistically Significant Detection of Semantic Shifts using Contextual Word Embeddings. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 104–113, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Daniel Loureiro, Aminette D'Souza, Areej Nasser Muhajab, Isabella A. White, Gabriel Wong, Luis Espinosa-Anke, Leonardo Neves, Francesco Barbieri, and Jose Camacho-Collados. 2022. TempoWiC:

An Evaluation Benchmark for Detecting Meaning Shift in Social Media. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3353–3359, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Brady D Lund and Ting Wang. 2023. Chatting about ChatGPT: How May AI and GPT Impact Academia and Libraries? *Library Hi Tech News*, 40(3):26–29.

Chenyang Lyu, Yongxin Zhou, and Tianbo Ji. 2022. MLLabs-LIG at TempoWiC 2022: A Generative Approach for Examining Temporal Meaning Shift. In *Proceedings of the The First Workshop on Ever Evolving NLP (EvoNLP)*, pages 1–6, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Xianghe Ma, Michael Strube, and Wei Zhao. 2024a. Graph-based Clustering for Detecting Semantic Change Across Time and Languages. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1542–1561, St. Julian's, Malta. Association for Computational Linguistics.

Xianghe Ma, Michael Strube, and Wei Zhao. 2024b. Graph-based Clustering for Detecting Semantic Change Across Time and Languages. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1542–1561, St. Julian's, Malta. Association for Computational Linguistics.

Xiaofei Ma, Zhiguo Wang, Patrick Ng, Ramesh Nallapati, and Bing Xiang. 2019. Universal Text Representation from BERT: An Empirical Study.

Ansel MacLaughlin, Shaobin Xu, and David A. Smith. 2021. Recovering Lexically and Semantically Reused Texts. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 52–66, Online. Association for Computational Linguistics.

James MacQueen et al. 1967. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Nikolay Malkin, Sameera Lanka, Pranav Goel, Sudha Rao, and Nebojsa Jojic. 2021. GPT Perdetry Test: Generating New Meanings for New Words. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5542–5553, Online. Association for Computational Linguistics.

Suresh Manandhar, Ioannis Klapaftis, Dmitriy Dligach, and Sameer Pradhan. 2010. SemEval-2010 Task 14: Word Sense Induction & Disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 63–68, Uppsala, Sweden. Association for Computational Linguistics.

Enrique Manjavacas, Brian Long, and Mike Kestemont. 2019. On the Feasibility of Automated Detection of Allusive Text Reuse. In *Proceedings of the 3rd Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 104–114, Minneapolis, USA. Association for Computational Linguistics.

Christopher D Manning. 2009. *An Introduction to Information Retrieval*. Cambridge university press.

Stratos Mansalis, Eirini Ntoutsi, Nikos Pelekis, and Yannis Theodoridis. 2018. An evaluation of data stream clustering algorithms. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 11(4):167–187.

Federico Martelli, Najla Kalach, Gabriele Tola, and Roberto Navigli. 2021. SemEval-2021 Task 2: Multilingual and Cross-lingual Word-in-Context Disambiguation (MCL-WiC). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 24–36, Online. Association for Computational Linguistics.

Matej Martinc, Petra Kralj Novak, and Senja Pollak. 2020a. Leveraging Contextual Embeddings for Detecting Diachronic Semantic Shift. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4811–4819, Marseille, France. European Language Resources Association.

Matej Martinc, Syrielle Montariol, Elaine Zosa, and Lidia Pivovarova. 2020b. Capturing Evolution in Word Usage: Just Add More Clusters? In *Companion Proceedings of the Web Conference 2020*, WWW '20, page 343–349, Taipei, Taiwan. Association for Computing Machinery.

Matej Martinc, Syrielle Montariol, Elaine Zosa, and Lidia Pivovarova. 2020c. Discovery Team at SemEval-2020 Task 1: Context-sensitive Embeddings Not Always Better than Static for Semantic Change Detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 67–73, Barcelona (online). International Committee for Computational Linguistics.

Terence E. McDonnell, Christopher A. Bail, and Iddo Tavory. 2017. A Theory of Resonance. *Sociological Theory*, 35(1):1–14.

Barbara McGillivray, Dominik Schlechtweg, Haim Dubossarsky, Nina Tahmasebi, and Simon Hengchen. 2021. DWUG LA: Diachronic Word Usage Graphs for Latin.

Leland McInnes, John Healy, and Steve Astels. 2017. HDBSCAN: Hierarchical Density Based Clustering. *Journal of Open Source Software*, 2(11):205.

Leland McInnes, John Healy, and James Melville. 2020. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction.

Qiaozhu Mei and ChengXiang Zhai. 2005. Discovering Evolutionary Theme Patterns from Text: An Exploration of Temporal Text Mining. In *Proceedings of the Eleventh ACM SIGKDD International Conference* *on Knowledge Discovery in Data Mining*, KDD '05, pages 198–207, Chicago, Illinois, USA. Association for Computing Machinery.

Stefano Menini, Teresa Paccosi, Sara Tonelli, Marieke Van Erp, Inger Leemans, Pasquale Lisena, Raphael Troncy, William Tullett, Ali Hürriyetoğlu, Ger Dijkstra, Femke Gordijn, Elias Jürgens, Josephine Koopman, Aron Ouwerkerk, Sanne Steen, Inna Novalija, Janez Brank, Dunja Mladenic, and Anja Zidar. 2022. A Multilingual Benchmark to Capture Olfactory Situations over Time. In *Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change*, pages 1–10, Dublin, Ireland. Association for Computational Linguistics.

Timothee Mickus, Denis Paperno, Mathieu Constant, and Kees van Deemter. 2020. What do you mean, BERT? In *Proceedings of the Society for Computation in Linguistics 2020*, pages 279–290, New York, New York. Association for Computational Linguistics.

Timothee Mickus, Denis Paperno, and Matthieu Constant. 2019. Mark my Word: A Sequence-to-Sequence Approach to Definition Modeling. In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, pages 1–11, Turku, Finland. Linköping University Electronic Press.

Timothee Mickus, Kees Van Deemter, Mathieu Constant, and Denis Paperno. 2022. Semeval-2022 Task 1: CODWOE – Comparing Dictionaries and Word Embeddings. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1–14, Seattle, United States. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Filip Miletic, Anne Przewozny-Desriaux, and Ludovic Tanguy. 2021. Detecting Contact-Induced Semantic Shifts: What Can Embedding-Based Methods Do in Practice? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10852–10865, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

George A. Miller. 1994. WordNet: A Lexical Database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

George A. Miller and Walter G. Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A Semantic Concordance. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Syrielle Montariol, Matej Martinc, and Lidia Pivovarova. 2021. Scalable and Interpretable Semantic Change Detection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4642–4652, Online. Association for Computational Linguistics.

Maria Moritz, Andreas Wiederhold, Barbara Pavlek, Yuri Bizzoni, and Marco Büchler. 2016. Non-Literal Text Reuse in Historical Texts: An Approach to Identify Reuse Transformations and its Application to Bible Reuse. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1849–1859, Austin, Texas. Association for Computational Linguistics.

Iqra Muneer and Rao Muhammad Adeel Nawab. 2022. Cross-Lingual Text Reuse Detection at sentence level for English–Urdu language pair. *Computer Speech & Language*, 75:101381.

Juan Pablo Munoz, Jinjie Yuan, Yi Zheng, and Nilesh Jain. 2024. LoNAS: Elastic Low-Rank Adapters for Efficient Large Language Models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10760–10776, Torino, Italia. ELRA and ICCL.

Usman Naseem, Imran Razzak, Shah Khalid Khan, and Mukesh Prasad. 2021. A Comprehensive Survey on Word Representation Models: From Classical to State-of-the-Art Word Representation Language Models. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 20(5).

D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz. 1998. UCI Repository of Machine Learning Databases.

Ke Ni and William Yang Wang. 2017. Learning to Explain Non-Standard English Words and Phrases. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 413–417, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Bill Noble, Asad Sayeed, Raquel Fernández, and Staffan Larsson. 2021. Semantic Shift in Social Networks. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 26–37, Online. Association for Computational Linguistics.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition Modeling: Learning to Define Word Embeddings in Natural Language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Ben O'Neill. 2011. A Critique of Politically Correct Language. *The Independent Review*, 16(2):279–291.

OpenAI. 2023. GPT-4 Technical Report.

Naho Orita, Naomi Feldman, Jordan Boyd-Graber, and Eliana Vornov. 2014. Quantifying the Role of Discourse Topicality in Speakers' Choices of Referring Expressions. In *Proceedings of the Fifth Workshop on Cognitive Modeling and Computational Linguistics*, pages 63–70, Baltimore, Maryland, USA. Association for Computational Linguistics.

Lionel Ott and Fabio Ramos. 2012. Unsupervised Incremental Learning for Long-term Autonomy. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4022–4029.

Teresa Paccosi, Stefano Menini, Elisa Leonardelli, Ilaria Barzon, and Sara Tonelli. 2023. Scent and Sensibility: Perception Shifts in the Olfactory Domain. In *Proceedings of the 4th Workshop on Computational Approaches to Historical Language Change*, pages 143–152, Singapore. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Jeong Yeon Park, Hyeong Jin Shin, and Jae Sung Lee. 2022. Word Sense Disambiguation Using Clustered Sense Labels. *Applied Sciences*, 12(4).

Hermann Paul. 1880. *Prinzipien der Sprachgeschichte*. Niemeyer, Halle.

Paolo Pedinotti and Alessandro Lenci. 2020. Don't Invite BERT to Drink a Bottle: Modeling the Interpretation of Metonymies Using BERT and Distributional Representations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6831–6837, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Fabian Pedregosa, Gäel Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Francesco Periti. 2023. Contextualised Semantic Shift Detection. In *Proceedings of the 31st Symposium of Advanced Database Systems (SEBD)*, pages 735–741, Galzingano Terme, Italy. CEUR.org.

Francesco Periti, David Alfter, and Nina Tahmasebi. 2024a. Automatically Generated Definitions and their utility for Modeling Word Meaning. In *Proceedings of the 2024 Conference on Empirical Methods in*

Natural Language Processing, pages 14008–14026, Miami, Florida, USA. Association for Computational Linguistics.

Francesco Periti, Pierluigi Cassotti, Haim Dubossarsky, and Nina Tahmasebi. 2024b. Analyzing Semantic Change through Lexical Replacements. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4495–4510, Bangkok, Thailand. Association for Computational Linguistics.

Francesco Periti, Pierluigi Cassotti, Stefano Montanelli, Nina Tahmasebi, and Dominik Schlechtweg. 2024c. TRoTR: A Framework for Evaluating the Re-contextualization of Text Reuse. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13972–13990, Miami, Florida, USA. Association for Computational Linguistics.

Francesco Periti and Haim Dubossarsky. 2023. The Time-Embedding Travelers@WiC-ITA. In *Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023)*, Parma, Italy. CEUR.org.

Francesco Periti, Haim Dubossarsky, and Nina Tahmasebi. 2024d. (Chat)GPT v BERT Dawn of Justice for Semantic Change Detection. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 420–436, St. Julian's, Malta. Association for Computational Linguistics.

Francesco Periti, Alfio Ferrara, Stefano Montanelli, and Martin Ruskov. 2022. What is Done is Done: an Incremental Approach to Semantic Shift Detection. In *Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change*, pages 33–43, Dublin, Ireland. Association for Computational Linguistics.

Francesco Periti and Stefano Montanelli. 2024. Lexical Semantic Change through Large Language Models: a Survey. *ACM Comput. Surv.*, 56(11).

Francesco Periti, Sergio Picascia, Stefano Montanelli, Alfio Ferrara, and Nina Tahmasebi. 2024e. Studying Word Meaning Evolution through Incremental Semantic Shift Detection. *Language Resources and Evaluation*.

Francesco Periti and Nina Tahmasebi. 2024a. A Systematic Comparison of Contextualized Word Embeddings for Lexical Semantic Change. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4262–4282, Mexico City, Mexico. Association for Computational Linguistics.

Francesco Periti and Nina Tahmasebi. 2024b. Towards a Complete Solution to Lexical Semantic Change: an Extension to Multiple Time Periods and Diachronic Word Sense Induction. In *Proceedings of the 5th Workshop on Computational Approaches to Historical Language Change*, pages 108–119, Bangkok, Thailand. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge Enhanced Contextual Word Representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Erika Petersen and Christopher Potts. 2023. Lexical Semantics with Large Language Models: A Case Study of English "break". In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 490–511, Dubrovnik, Croatia. Association for Computational Linguistics.

Kevin J Peterson and Hongfang Liu. 2021. An Examination of the Statistical Laws of Semantic Change in Clinical Notes. *AMIA Joint Summits on Translational Science proceedings*, 2021:515–524.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the Word-in-Context Dataset for Evaluating Context-Sensitive Meaning Representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2021. *Contextualized Embeddings*, pages 69–96. Springer International Publishing, Cham.

Martin Pömsl and Roman Lyapin. 2020. CIRCE at SemEval-2020 Task 1: Ensembling Context-Free and Context-Dependent Word Representations. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 180–186, Barcelona (online). International Committee for Computational Linguistics.

Jack A. Porrino, Virak Tan, and Aaron Daluiski. 2008. Misquotation of a Commonly Referenced Hand Surgery Study. *The Journal of Hand Surgery*, 33(1):2.e1–2.e9.

Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Forough Poursabzi-Sangdeh and Jordan Boyd-Graber. 2015. Speeding Document Annotation with Topic Models. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 126–132, Denver, Colorado. Association for Computational Linguistics.

Marko Pranjić, Kaja Dobrovoljc, Senja Pollak, and Matej Martinc. 2024. Semantic Change Detection for Slovene Language: a Novel Dataset and an Approach Based on Optimal Transport.

Judita Preiss. 2024. Using Word Evolution to Predict Drug Repurposing. *BMC Medical Informatics and Decision Making*, 24(2):114.

James Pustejovsky and Branimir Boguraev. 1993. Lexical Knowledge Representation and Natural Language Processing. *Artificial Intelligence*, 63(1):193–223.

Wenjun Qiu and Xu Yang. 2022. HistBERT: A Pre-trained Language Model for Diachronic Lexical Semantic Analysis.

Maxim Rachinskiy and Nikolay Arefyev. 2021. Zeroshot Crosslingual Transfer of a Gloss Language Model for Semantic Change Detection. In *Proceedings of the Conference on Computational Linguistics and Intellectual Technologies (Dialogue)*, (online). RSUH.

Maxim Rachinskiy and Nikolay Arefyev. 2022. GlossReader at LSCDiscovery: Train to Select a Proper Gloss in English – Discover Lexical Semantic Change in Spanish. In *Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change*, pages 198–203, Dublin, Ireland. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Alessandro Raganato, Tommaso Pasini, Jose Camacho-Collados, and Mohammad Taher Pilehvar. 2020. XL-WiC: A Multilingual Benchmark for Evaluating Semantic Contextualization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7193–7206, Online. Association for Computational Linguistics.

William M. Rand. 1971. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, 66(336):846–850.

Abhilasha Ravichander, Eduard Hovy, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung. 2020. On the Systematicity of Probing Contextualized Word Representations: The Case of Hypernymy in BERT. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 88–102, Barcelona, Spain (Online). Association for Computational Linguistics.

Radim Rehurek and Petr Sojka. 2011. Gensim–Python Framework for Vector Space Modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2).

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and Measuring the Geometry of BERT. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Karl Christian Reisig. 1839. *Professor K. Reisig's Vorlesungen Über Lateinische Sprachwissenschaft*. Verlag der Lehnhold'schen Buchhandlung.

Frederick Riemenschneider and Anette Frank. 2023. Exploring Large Language Models for Classical Philology. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15181–15199, Toronto, Canada. Association for Computational Linguistics.

Shruti Rijhwani and Daniel Preotiuc-Pietro. 2020. Temporally-Informed Analysis of Named Entity Recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7605–7617, Online. Association for Computational Linguistics.

Julia Rodina, Yuliya Trofimova, Andrey Kutuzov, and Ekaterina Artemova. 2021. ELMo and BERT in Semantic Change Detection for Russian. In *Analysis of Images, Social Networks and Texts*, pages 175–186, Cham. Springer International Publishing.

Guy D. Rosin, Ido Guy, and Kira Radinsky. 2022. Time Masking for Temporal Language Models. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, page 833–841, Virtual Event, AZ, USA. Association for Computing Machinery.

Guy D. Rosin and Kira Radinsky. 2022. Temporal Attention for Language Models. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1498–1508, Seattle, United States. Association for Computational Linguistics.

David Rother, Thomas Haider, and Steffen Eger. 2020. CMCE at SemEval-2020 Task 1: Clustering on Manifolds of Contextualized Embeddings to Detect Historical Meaning Shifts. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 187–193, Barcelona (online). International Committee for Computational Linguistics.

Paul Röttger and Janet Pierrehumbert. 2021. Temporal Adaptation of BERT and Performance on Downstream Document Classification: Insights from Social Media. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2400–2412, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Peter J. Rousseeuw. 1987. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Maja Rudolph and David Blei. 2018. Dynamic Embeddings for Language Evolution. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 1003–1011, Lyon, France. International World Wide Web Conferences Steering Committee.

Anastasiia Ryzhova, Daria Ryzhova, and Ilya Sochenkov. 2021. Detection of Semantic Changes in Russian Nouns with Distributional Models and Grammatical Features. In *Proceedings of the Conference on Computational Linguistics and Intellectual Technologies (Dialogue)*, (online). RSUH.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter.

Yo Sato and Kevin Heffernan. 2020. Homonym normalisation by word sense clustering: a case in Japanese. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3324–3332, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Dominik Schlechtweg. 2023. *Human and Computational Measurement of Lexical Semantic Change*. Ph.D. thesis, University of Stuttgart.

Dominik Schlechtweg, Haim Dubossarsky, Simon Hengchen, Barbara McGillivray, and Nina Tahmasebi. 2022a. DWUG EN: Diachronic Word Usage Graphs for English.

Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. 2020. SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1–23, Barcelona (online). International Committee for Computational Linguistics.

Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. 2022b. DWUG DE: Diachronic Word Usage Graphs for German.

Dominik Schlechtweg, Nina Tahmasebi, Simon Hengchen, Haim Dubossarsky, and Barbara McGillivray. 2021. DWUG: A large Resource of Diachronic Word Usage Graphs in Four Languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7079–7091, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dominik Schlechtweg, Shafqat Mumtaz Virk, Pauline Sander, Emma Sköldberg, Lukas Theuer Linke, Tuo Zhang, Nina Tahmasebi, Jonas Kuhn, and Sabine Schulte im Walde. 2024. The DURel Annotation Tool: Human and Computational Measurement of Semantic Proximity, Sense Clusters and Semantic Change. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 137–149, St. Julians, Malta. Association for Computational Linguistics.

Dominik Schlechtweg and Sabine Schulte im Walde. 2020. Simulating Lexical Semantic Change from Sense-Annotated Data. In *Proceedings of the 13th International Conference on the Evolution of Language (EvoLang13)*, Brussels, Belgium.

Dominik Schlechtweg, Sabine Schulte im Walde, and Stefanie Eckmann. 2018. Diachronic Usage Relatedness (DURel): A Framework for the Annotation of Lexical Semantic Change. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 169–174, New Orleans, Louisiana. Association for Computational Linguistics.

Hinrich Schütze. 1998. Automatic Word Sense Discrimination. *Computational Linguistics*, 24(1):97–123.

Vincent Segonne and Timothee Mickus. 2023. Definition Modeling : To model definitions. Generating Definitions With Little to No Semantics. In *Proceedings of the 15th International Conference on Computational Semantics*, pages 258–266, Nancy, France. Association for Computational Linguistics.

Frank Seifart. 2019. *Contact-induced Change*. De Gruyter Mouton, Berlin, Boston.

Lutfi Kerem Senel, Ihsan Utlu, Veysel Yucesoy, Aykut Koc, and Tolga Cukur. 2018. Semantic Structure and Interpretability of Word Embeddings. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 26(10):1769–1779.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Jangwon Seo and W. Bruce Croft. 2008. Local Text Reuse Detection. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, page 571–578, New York, NY, USA. Association for Computing Machinery.

Sofia Serrano and Noah A. Smith. 2019. Is Attention Interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face.

X. H. Shi, R. C. Guan, L. P. Wang, Z. L. Pei, and Y. C. Liang. 2009. An incremental affinity propagation algorithm and its applications for text clustering. In *Proceedings of the 2009 International Joint Conference on Neural Networks*, IJCNN'09, pages 2734–2739, Atlanta, Georgia, USA. IEEE Press.

Philippa Shoemark, Farhana Ferdousi Liza, Dong Nguyen, Scott Hale, and Barbara McGillivray. 2019. Room to Glo: A Systematic Comparison of Semantic Change Detection Approaches with Word Embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 66–76, Hong Kong, China. Association for Computational Linguistics.

Jonathan A. Silva, Elaine R. Faria, Rodrigo C. Barros, Eduardo R. Hruschka, André C. P. L. F. de Carvalho, and João Gama. 2013. Data stream clustering: A survey. *ACM Comput. Surv.*, 46(1).

David A. Smith, Ryan Cordel, Elizabeth Maddock Dillon, Nick Stramp, and John Wilkerson. 2014. Detecting and Modeling Local Text Reuse. In *IEEE/ACM Joint Conference on Digital Libraries*, pages 183–192.

David A. Smith, Ryan Cordell, and Elizabeth Maddock Dillon. 2013. Infectious Texts: Modeling Text Reuse in Nineteenth-Century Newspapers. In *2013 IEEE International Conference on Big Data*, pages 86–94.

Padhraic Smyth. 1996. Clustering Sequences with Hidden Markov Models. In *Advances in Neural Information Processing Systems*, volume 9. MIT Press.

C. Spearman. 1987. The Proof and Measurement of Association between Two Things. *The American Journal of Psychology*, 100(3/4):441–471.

Gustaf Stern. 1931. *Meaning and Change of Meaning; with Special Reference to the English Language*. Wettergren & Kerbers.

Zhaochen Su, Zecheng Tang, Xinyan Guan, Lijun Wu, Min Zhang, and Juntao Li. 2022. Improving Temporal Generalization of Pre-trained Language Models with Lexical Semantic Change. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6380–6393, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Leilei Sun and Chonghui Guo. 2014. Incremental Affinity Propagation Clustering Based on Message Passing. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 26(11):2731–2744.

Aris Sunmood, Thanawin Rakthanmanon, and Kitsana Waiyamai. 2018. Evolution and Affinity-Propagation Based Approach for Data Stream Clustering. In *Proceedings of the International Conference on Frontiers of Educational Technologies (ICFET)*, pages 97–101.

Jader Martins Camboim de Sá, Marcos Da Silveira, and Cédric Pruski. 2024. Survey in Characterization of Semantic Change.

Nina Tahmasebi, Lars Borin, and Adam Jatowt. 2021a. *Survey of Computational Approaches to Lexical Semantic Change Detection*, pages 1–91. Language Science Press, Berlin.

Nina Tahmasebi, Lars Borin, Adam Jatowt, and Yang Xu, editors. 2019. *Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change*. Association for Computational Linguistics, Florence, Italy.

Nina Tahmasebi and Haim Dubossarsky. 2023. Computational Modeling of Semantic Change.

Nina Tahmasebi, Simon Hengchen, Dominik Schlechtweg, Barbara McGillivray, and Haim Dubossarsky. 2022a. DWUG SV: Diachronic Word Usage Graphs for Swedish.

Nina Tahmasebi, Adam Jatowt, Yang Xu, Simon Hengchen, Syrielle Montariol, and Haim Dubossarsky, editors. 2021b. *Proceedings of the 2nd International Workshop on Computational Approaches to Historical Language Change 2021*. Association for Computational Linguistics, Online.

Nina Tahmasebi, Syrielle Montariol, , Andrey Kutuzov, David Alfter, Francesco Periti, Pierluigi Cassotti, and Netta Huebscher, editors. 2024. *Proceedings of the 5th Workshop on Computational Approaches to Historical Language Change*. Association for Computational Linguistics, Bangkok.

Nina Tahmasebi, Syrielle Montariol, Haim Dubossarsky, Andrey Kutuzov, Simon Hengchen, David Alfter, Francesco Periti, and Pierluigi Cassotti, editors. 2023. *Proceedings of the 4th Workshop on Computational Approaches to Historical Language Change*. Association for Computational Linguistics, Singapore.

Nina Tahmasebi, Syrielle Montariol, Andrey Kutuzov, Simon Hengchen, Haim Dubossarsky, and Lars Borin, editors. 2022b. *Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change*. Association for Computational Linguistics, Dublin, Ireland.

Nina Tahmasebi, Syrielle Montariol, Andrey Kutuzov, Simon Hengchen, Haim Dubossarsky, and Lars Borin, editors. 2022c. *Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change*. Association for Computational Linguistics, Dublin, Ireland.

Nina Tahmasebi, Kai Niklas, Gideon Zenz, and Thomas Risse. 2013. On the Applicability of Word Sense Discrimination on 201 Years of Modern English. *International Journal on Digital Libraries*, 13(3-4):135–153.

Nina Tahmasebi and Thomas Risse. 2017. Finding Individual Word Sense Changes and their Delay in Appearance. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 741–749, Varna, Bulgaria. INCOMA Ltd.

Xiaohang Tang, Yi Zhou, Taichi Aida, Procheta Sen, and Danushka Bollegala. 2023. Can Word Sense Distribution Detect Semantic Changes of Words? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3575–3590, Singapore. Association for Computational Linguistics.

Xuri Tang. 2018. A State-of-the-art of Semantic Change Computation. *Natural Language Engineering*, 24(5):649–676.

Daniela Teodorescu, Spencer von der Ohe, and Grzegorz Kondrak. 2022. UAlberta at LSCDiscovery: Lexical Semantic Change Detection via Word Sense Disambiguation. In *Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change*, pages 180–186, Dublin, Ireland. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 296–310, Online. Association for Computational Linguistics.

Martyn P. Thompson. 1993. Reception Theory and the Interpretation of Historical Meaning. *History and Theory*, 32(3):248–272.

Konstantin Todorov and Giovanni Colavizza. 2022. An Assessment of the Impact of OCR Noise on Language Models.

Amirsina Torfi, Rouzbeh A. Shirvani, Yaser Keneshloo, Nader Tavaf, and Edward A. Fox. 2021. Natural Language Processing Advancements By Deep Learning: A Survey.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and Efficient Foundation Language Models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open Foundation and Fine-Tuned Chat Models.

Yu-Hsiang Tseng, Mao-Chang Ku, Wei-Ling Chen, Yu-Lin Chang, and Shu-Kai Hsieh. 2023. Vec2Gloss: definition modeling leveraging contextualized vectors with Wordnet gloss. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 679–690.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-Read Students Learn Better: On the Importance of Pre-training Compact Models.

S. Ullmann. 1957. *The Principles of Semantics*. Glasgow University publications. Jackson.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Kitsana Waiyamai, Thanapat Kangkachit, Thanawin Rakthanmanon, and Rattanapong Chairukwattana. 2014. SED-Stream: Discriminative Dimension Selection for Evolution-Based Clustering of High Dimensional Data Streams. *International Journal of Intelligent Systems Technologies and Applications (IJISTA)*, 13(3):187–201.

Jonas Wallat, Fabian Beringer, Abhijit Anand, and Avishek Anand. 2023. Probing BERT for Ranking Abilities. In *Advances in Information Retrieval*, pages 255–273, Cham. Springer Nature Switzerland.

Benyou Wang, Emanuele Di Buccio, and Massimo Melucci. 2020. University of Padova @ DIACR-Ita. In *Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, Marrakech, Morocco. CEUR-WS.

Benyou Wang, Lifeng Shang, Christina Lioma, Xin Jiang, Hao Yang, Qun Liu, and Jakob Grue Simonsen. 2021a. On Position Embeddings in BERT. In *International Conference on Learning Representations*.

Lingzhi Wang, Xingshan Zeng, and Kam-Fai Wong. 2021b. Quotation Recommendation and Interpretation Based on Transformation from Queries to Quotations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 754–758, Online. Association for Computational Linguistics.

Lingzhi Wang, Xingshan Zeng, and Kam-Fai Wong. 2022. Learning When and What to Quote: A Quotation Recommender System with Mutual Promotion of Recommendation and Generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3094–3105, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Lingzhi Wang, Xingshan Zeng, and Kam-Fai Wong. 2023. Quotation Recommendation for Multi-Party Online Conversations Based on Semantic and Topic Fusion. *ACM Trans. Inf. Syst.*, 41(4).

Anna Wegmann, Florian Lemmerich, and Markus Strohmaier. 2020. Detecting Different Forms of Semantic Shift in Word Embeddings via Paradigmatic and Syntagmatic Association Changes. In *The Semantic Web – ISWC 2020*, pages 619–635, Cham. Springer International Publishing.

Hendryk Weiland, Maike Behrendt, and Stefan Harmeling. 2023. Automatic Dictionary Generation: Could Brothers Grimm Create a Dictionary with BERT? In *Proceedings of the 19th Conference on Natural Language Processing (KONVENS 2023)*, pages 102–120, Ingolstadt, Germany. Association for Computational Lingustics.

Leonie Weissweiler, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Haofei Yu, Hinrich Schuetze, Kemal Oflazer, and David Mortensen. 2023. Counting the Bugs in ChatGPT's Wugs: A Multilingual Investigation into the Morphological Capabilities of a Large Language Model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6508–6524, Singapore. Association for Computational Linguistics.

Laura Wendlandt, Jonathan K. Kummerfeld, and Rada Mihalcea. 2018. Factors Influencing the Surprising Instability of Word Embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2092–2102, New Orleans, Louisiana. Association for Computational Linguistics.

Sean Wilner, Daniel Woolridge, and Madeleine Glick. 2021. Narrative Embedding: Re-Contextualization Through Attention. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1405, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation.

Katherine Wysocki and Joseph R. Jenkins. 1987. Deriving Word Meanings through Morphological Generalization. *Reading Research Quarterly*, 22(1):66–81.

Yu Xiao, Naomi Baes, Ekaterina Vylomova, and Nick Haslam. 2023. Have the Concepts of 'anxiety' and 'depression' been Normalized or Pathologized? A Corpus Study of Historical Semantic Change. *PloS One*, 18(6):e0288027.

Shaobin Xu, David Smith, Abigail Mullen, and Ryan Cordell. 2014. Detecting and Evaluating Local Text Reuse in Social Networks. In *Proceedings of the Joint Workshop on Social Dynamics and Personal Attributes in Social Media*, pages 50–57, Baltimore, Maryland. Association for Computational Linguistics.

Yang Xu and Charles Kemp. 2015. A Computational Evaluation of Two Laws of Semantic Change. In *Proceedings of the 37th Annual Meeting of the Cognitive Science Society*, Pasadena, California, USA. Cognitive Science Society.

Erjia Yan and Yongjun Zhu. 2018. Tracking Word Semantic Change in Biomedical Literature. *International Journal of Medical Informatics*, 109:76–86.

Chen Yang, Lorenzo Bruzzone, Renchu Guan, Laijun Lu, and Yanchun Liang. 2013. Incremental and Decremental Affinity Propagation for Semisupervised Clustering in Multispectral Images. *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, 51(3):1666–1679.

David Yenicelik, Florian Schmidt, and Yannic Kilcher. 2020. How does BERT Capture Semantics? A Closer Look at Polysemous Words. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 156–162, Online. Association for Computational Linguistics.

Jiaxin Yuan, Cunliang Kong, Chenhui Xie, Liner Yang, and Erhong Yang. 2022. COMPILING: A Benchmark Dataset for Chinese Complexity Controllable Definition Generation. In *Proceedings of the 21st Chinese National Conference on Computational Linguistics*, pages 921–931, Nanchang, China. Chinese Information Processing Society of China.

Frank D. Zamora-Reina, Felipe Bravo-Marquez, and Dominik Schlechtweg. 2022a. DWUG ES: Diachronic Word Usage Graphs for Spanish.

Frank D. Zamora-Reina, Felipe Bravo-Marquez, and Dominik Schlechtweg. 2022b. LSCDiscovery: A Shared Task on Semantic Change Discovery and Detection in Spanish. In *Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change*, pages 149–164, Dublin, Ireland. Association for Computational Linguistics.

Ziqian Zeng, Xin Liu, and Yangqiu Song. 2018. Biased Random Walk based Social Regularization for Word Embeddings. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4560–4566. International Joint Conferences on Artificial Intelligence Organization.

Hengyuan Zhang, Dawei Li, Yanran Li, Chenming Shang, Chufan Shi, and Yong Jiang. 2023. Assisting Language Learners: Automated Trans-Lingual Definition Generation via Contrastive Prompt Learning. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 260–274, Toronto, Canada. Association for Computational Linguistics.

Hengyuan Zhang, Dawei Li, Shiping Yang, and Yanran Li. 2022. Fine-grained Contrastive Learning for Definition Generation. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1001–1012, Online only. Association for Computational Linguistics.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2024. Instruction Tuning for Large Language Models: A Survey.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT.

Xiangliang Zhang, Cyril Furtlehner, and Michèle Sebag. 2008. Frugal and Online Affinity Propagation. In *Proceedings of the Conférence francophone sur l'Apprentissage (CAP)*.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2023. Can ChatGPT Understand Too? A Comparative Study on ChatGPT and Fine-tuned BERT.

Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, Hao Peng, Jianxin Li, Jia Wu, Ziwei Liu, Pengtao Xie, Caiming Xiong, Jian Pei, Philip S. Yu, and Lichao Sun. 2023a. A Comprehensive Survey on Pretrained Foundation Models: A History from BERT to ChatGPT.

Jinan Zhou and Jiaxin Li. 2020. TemporalTeller at SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection with Temporal Referencing. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 222–231, Barcelona (online). International Committee for Computational Linguistics.

Kaitlyn Zhou, Kawin Ethayarajh, and Dan Jurafsky. 2021. Frequency-based Distortions in Contextualized Word Embeddings.

Wei Zhou, Nina Tahmasebi, and Haim Dubossarsky. 2023b. The Finer They Get: Combining Fine-Tuned Models For Better Semantic Change Detection. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 518–528, Tórshavn, Faroe Islands. University of Tartu Library.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27.

Chapter A

A very first evaluation of ChatGPT

This appendix contains material for Chapter 3. For each considered temperature, we conducted two experiments. The comprehensive ChatGPT API results for Experiment 1 and Experiment 2 at different temperatures are presented in Tables A.1 and A.2. The average results of these two experiments are summarized in Table A.3.

		Experiment 1: ChatGPT API performance (Macro-F1) per temperature (0.0-2.0)											
	prompt	*0.0*	*0.2*	*0.4*	*0.6*	*0.8*	*1.0*	*1.2*	*1.4*	*1.6*	*1.8*	*2.0*	**avg**
TempoWiC	*ZSp*	.568	.584	.604	.599	.592	.576	.604	.560	.560	.599	.579	.584
	FSp	.648	.648	.664	.634	.597	.631	.645	.585	.608	.581	.598	.622
HistoWiC	*ZSp*	.728	.683	.689	.676	.666	.694	.715	.609	.704	.671	.594	.675
	FSp	.684	.698	.721	.698	.671	.700	.686	.599	.552	.607	.601	.656

Table A.1: Experiment 1: ChatGPT API performance (Macro-F1) for TempoWiC and HistoWiC.

		Experiment 2: ChatGPT API performance (Macro-F1) per temperature (0.0-2.0)											
	prompt	*0.0*	*0.2*	*0.4*	*0.6*	*0.8*	*1.0*	*1.2*	*1.4*	*1.6*	*1.8*	*2.0*	**avg**
TempoWiC	*ZSp*	.645	.628	.643	.605	.664	.602	.600	.598	.575	.580	.636	.616
	FSp	.659	.632	.649	.627	.644	.597	.689	.627	.597	.551	.562	.621
HistoWiC	*ZSp*	.751	.758	.711	.765	.729	.712	.678	.652	.679	.664	.604	.700
	FSp	.684	.678	.707	.700	.706	.665	.607	.662	.615	.592	.623	.658

Table A.2: Experiment 2: ChatGPT performance (Macro-F1) for TempoWiC and HistoWiC.

		Average: ChatGPT API performance (Macro-F1) per temperature (0.0-2.0)											
	prompt	*0.0*	*0.2*	*0.4*	*0.6*	*0.8*	*1.0*	*1.2*	*1.4*	*1.6*	*1.8*	*2.0*	**avg**
TempoWiC	*ZSp*	.606	.606	.624	.602	.628	.589	.602	.579	.568	.589	.607	.600
	FSp	.654	.640	.657	.631	.620	.614	.667	.606	.602	.566	.580	.622
HistoWiC	*ZSp*	.740	.720	.700	.720	.698	.703	.696	.631	.692	.668	.599	.688
	FSp	.684	.688	.714	.699	.688	.682	.647	.631	.584	.599	.612	.657

Table A.3: Average of experiment 1 and 2: ChatGPT API performance (Macro-F1) for TempoWiC and HistoWiC. We report the average performance for each temperature.

Chapter B

A systematic evaluation of word embeddings

This appendix contains material for Chapter 7.

B.1 Comprehensive evaluation

We report in TableB.1 a comprehensive evaluation of standard approaches to GCD by using the layers 1-12 of BERT / mBERT / XLM-R.

			EN	LA	DE	SV	ES	RU			NO		ZH	Avg$_w$
			$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_2 - C_3$	$C_1 - C_3$	$C_1 - C_2$	$C_2 - C_3$	$C_1 - C_2$	$C_i - C_j$
form-based	APD	1	.358 / .278 / .064	- / **.153** / .073	.144 / .218 / .270	.213 / .132 / .134	.167 / .104 / .003	.335 / .204 / .258	.281 / .204 / .308	.261 / .214 / .253	.160 / .143 / .145	.234 / .219 / .203	.340 / -.100 / -.222	.255 / .171 / .166
		2	.464 / .346 / .229	- / .119 / .006	.155 / .208 / .319	.255 / .129 / .234	.255 / .164 / .076	.374 / .198 / .245	.309 / .188 / .283	.303 / .218 / .236	.199 / .155 / .153	.288 / .213 / .235	.540 / .263 / .338	.312 / .198 / .216
		3	.574 / .389 / .314	- / .047 / -.025	.164 / .232 / .301	.295 / .189 / .289	.307 / .212 / .139	.427 / .215 / .238	.370 / .218 / .292	.360 / .242 / .241	.290 / .170 / .171	.371 / .223 / .243	.594 / .464 / .540	.371 / .232 / .244
		4	.628 / .410 / .400	- / .022 / -.010	.176 / .241 / .326	.307 / .254 / .286	.394 / .276 / .184	.492 / .257 / .287	.427 / .247 / .346	.431 / .280 / .288	.364 / .168 / .143	.463 / .322 / .264	**.747** / .613 / .615	.438 / .275 / .284
		5	**.684** / .412 / .452	- / -.028 / .043	.237 / .344 / .414	**.305** / .321 / .351	.450 / .345 / .279	.519 / .295 / .374	.465 / .275 / .453	.456 / .318 / .373	.396 / .192 / .165	.497 / .364 / .330	.720 / .662 / .600	.471 / .315 / .36
		6	.667 / .395 / .438	- / -.005 / .061	.309 / .397 / .471	.242 / .352 / .424	.468 / .361 / .277	.516 / .338 / .438	.463 / .305 / **.503**	.467 / .347 / **.432**	.400 / .180 / .172	.532 / .374 / .367	.667 / .661 / .629	.473 / .338 / **.398**
		7	.614 / **.419** / .395	- / -.009 / .073	.335 / .434 / .471	.237 / .404 / .441	**.479** / .364 / .280	.549 / .402 / **.439**	**.495** / .379 / .473	.523 / .429 / .430	.429 / .262 / .191	**.547** / .437 / .375	.645 / **.725** / .618	.494 / .390 / .393
		8	.642 / .408 / .426	- / .023 / .043	.389 / **.481** / .474	.248 / .455 / .456	.438 / **.430** / .297	**.566** / **.427** / .430	**.495** / **.400** / .466	.531 / **.451** / .427	.416 / **.291** / .197	.529 / **.499** / .373	.654 / .715 / .638	**.497** / **.421** / .396
		9	.600 / .406 / .460	- / .044 / -.047	**.427** / .423 / **.479**	.250 / **.463** / .468	.399 / .413 / .352	.539 / .382 / .401	.479 / .364 / .419	**.534** / .405 / .404	.429 / .257 / .190	.525 / .462 / .394	.667 / .670 / **.646**	.486 / .391 / .388
		10	.530 / .348 / .511	- / .008 / -.082	.354 / .333 / .433	.275 / .414 / .497	.282 / .331 / .407	.515 / .362 / .369	.461 / .313 / .405	.523 / .379 / .402	.418 / .226 / .191	.531 / .425 / .411	.625 / .656 / .613	.450 / .346 / .387
		11	.554 / .305 / **.548**	- / .023 / -.069	.275 / .315 / .409	.267 / .309 / **.500**	.257 / .265 / **.444**	.439 / .333 / .361	.393 / .256 / .394	.461 / .330 / .401	.378 / .196 / **.215**	.530 / .403 / **.432**	.604 / .628 / .601	.405 / .303 / .392
		12	.563 / .363 / .444	- / .102 / **.151**	.271 / .398 / .264	.270 / .389 / .257	.335 / .341 / .386	.518 / .368 / .290	.482 / .345 / .287	.476 / .386 / .318	**.441** / .279 / .195	.466 / .488 / .379	.656 / .689 / .500	.449 / .371 / .316
	PRT	1	.295 / .195 / .221	- / .289 / .303	.133 / .162 / .122	.215 / .001 / .045	.303 / .295 / .190	.263 / .271 / .220	.206 / .149 / .305	.159 / .169 / .144	.032 / -.005 / .028	.161 / .168 / .039	.383 / .017 / -.139	.220 / .178 / .165
		2	.409 / .271 / .382	- / .286 / .263	.217 / .198 / .125	.274 / .006 / .066	.407 / .397 / .328	.304 / .279 / .216	.261 / .139 / .352	.196 / .161 / .153	.122 / -.020 / .092	.349 / .215 / -.020	.582 / .192 / .140	.302 / .209 / .216
		3	.436 / .295 / .453	- / .277 / .271	.267 / .230 / .141	**.301** / .012 / .078	.438 / .424 / .364	.338 / .311 / .203	.305 / .191 / .405	.251 / .195 / .162	.250 / .042 / .111	.365 / .294 / .005	.676 / .397 / .424	.348 / .253 / .253
		4	.467 / .290 / .487	- / .255 / .297	.297 / .285 / .204	.280 / .017 / .087	.455 / .446 / .388	.398 / .329 / .246	.346 / .235 / .433	.306 / .250 / .234	.378 / .019 / .102	.408 / .303 / .075	.691 / .525 / .544	.389 / .283 / .296
		5	.494 / .315 / .476	- / .232 / .322	.343 / .384 / .294	.233 / .060 / .129	.455 / .495 / .439	.399 / .364 / .323	.395 / .327 / .509	.331 / .313 / .323	.440 / .096 / .137	.466 / .367 / .189	.651 / .551 / .531	.408 / .337 / .357
		6	.516 / .353 / .447	- / .257 / .350	.379 / .421 / .357	.206 / .082 / .171	.451 / .524 / .449	.391 / .359 / .365	.390 / .374 / .519	.331 / .365 / .384	.449 / .104 / .181	.471 / .330 / .232	.637 / .556 / .475	.408 / .362 / .383
		7	.529 / **.383** / .462	- / .304 / .349	.400 / .437 / .385	.178 / .008 / .184	.466 / .498 / .453	**.411** / .379 / .358	**.426** / **.447** / .510	**.380** / .413 / .384	**.511** / .161 / .192	.501 / .371 / .236	.641 / .613 / .549	**.433** / .389 / .390
		8	.539 / **.383** / .464	- / .292 / .359	.398 / .468 / .402	.197 / .081 / .196	.453 / .514 / .463	.404 / **.393** / .375	.410 / .421 / **.531**	**.380** / .411 / .396	.449 / .227 / .292	.493 / .389 / **.246**	.664 / **.619** / .575	.426 / **.400** / .409
		9	**.549** / .358 / .437	- / .311 / .319	.390 / **.469** / .477	.201 / .096 / .247	**.476** / .501 / .503	.375 / .353 / .382	.402 / .404 / .471	.353 / .384 / .401	.481 / **.243** / .351	.485 / .380 / .239	.671 / .606 / **.646**	.422 / .385 / .418
		10	.511 / .355 / .481	- / .280 / .329	.380 / .454 / .486	.193 / **.133** / .223	.417 / .482 / .538	.349 / .376 / .409	.379 / .382 / .447	.335 / .366 / .431	.482 / .212 / .373	.481 / .398 / .263	.626 / .583 / .619	.396 / .378 / .431
		11	.452 / .342 / **.501**	- / .298 / .308	.412 / .430 / **.507**	.169 / .076 / **.245**	.422 / .489 / **.540**	.319 / .344 / **.412**	.317 / .335 / .439	.303 / .321 / **.438**	.448 / .197 / .360	**.503** / .365 / .214	.602 / .550 / .620	.371 / .350 / **.432**
		12	.457 / .270 / .411	- / **.380** / **.424**	**.422** / .436 / .369	.158 / .193 / .020	.413 / **.543** / .505	.400 / .391 / .321	.374 / .356 / .443	.347 / **.423** / .405	.507 / .219 / **.387**	.444 / **.438** / .149	**.712** / .524 / .558	.406 / .395 / .381
sense-based	AP	1	.129 / .220 / .032	- / -.011 / **.409**	-.108 / -.087 / -.040	-.121 / -.021 / -.244	.168 / .233 / .172	.050 / -.001 / -.154	.132 / .108 / .060	.098 / -.143 / .023	-.104 / -.237 / -.019	-.048 / .021 / -.239	.118 / -.179 / .110	.060 / .011 / .012
		2	.288 / .079 / -.128	- / .008 / .215	.113 / -.131 / -.017	-.138 / -.141 / -.244	.104 / .109 / .140	-.127 / -.154 / -.036	.038 / .110 / .073	.096 / -.109 / -.025	.031 / -.230 / -.025	-.039 / .104 / .028	.301 / -.058 / -.048	.052 / -.030 / .006
		3	.267 / .161 / .016	- / -.012 / .218	.007 / -.043 / .120	-.201 / -.117 / -.177	.161 / .142 / .063	-.006 / .007 / -.019	-.002 / .058 / .129	.027 / -.130 / -.020	-.118 / .016 / -.060	-.051 / -.011 / .124	.189 / .221 / -.143	.033 / .021 / .028
		4	.353 / .330 / .087	- / -.106 / .253	-.041 / .088 / .054	-.213 / -.131 / -.172	.263 / .195 / **.266**	.093 / -.159 / -.042	.045 / .096 / .104	.168 / -.076 / .050	-.281 / -.123 / -.016	.257 / -.282 / .020	.360 / .322 / -.047	.113 / .014 / .064
		5	**.432** / .221 / .322	- / -.024 / .281	.143 / .235 / .196	-.015 / -.083 / -.125	.247 / .319 / .162	.072 / -.085 / -.035	.169 / .014 / .140	.081 / -.019 / .025	-.318 / -.027 / .033	.323 / .143 / .149	.251 / **.689** / .343	.140 / .097 / .112
		6	.431 / .208 / .330	- / -.000 / .286	.243 / .372 / .280	-.129 / -.040 / -.070	.363 / .251 / .002	-.049 / -.111 / -.094	.173 / .093 / .176	.091 / .035 / .291	-.192 / -.076 / .031	**.440** / .206 / .131	**.458** / .342 / .280	.166 / .099 / .132
		7	.144 / .362 / .321	- / -.044 / .233	.284 / **.443** / .387	-.070 / -.031 / -.155	**.406** / .301 / .216	.082 / -.069 / **.067**	.288 / **.235** / .084	**.190** / **.158** / .131	-.257 / -.114 / -.051	.115 / .140 / -.130	.292 / .226 / .344	.183 / .153 / .131
		8	.228 / **.418** / .175	- / -.101 / .260	.417 / .353 / .393	**.124** / .114 / -.082	.384 / **.401** / .031	.058 / -.014 / -.073	.128 / .230 / .211	.088 / .137 / .228	-.165 / -.114 / -.109	-.029 / **.469** / **.256**	.113 / .231 / .045	.148 / **.192** / .117
		9	.424 / .357 / .311	- / .120 / .153	.339 / .322 / .361	.054 / .010 / -.195	.270 / .296 / .157	.038 / .013 / -.081	.072 / .149 / .232	.098 / .055 / .011	-.016 / .005 / **.045**	.092 / .198 / .031	.423 / .404 / .245	.157 / .158 / .104
		10	.233 / .317 / .289	- / .124 / .381	.393 / .328 / .334	-.023 / .061 / -.210	.294 / .201 / .151	**.126** / **.108** / .044	.116 / .169 / .240	.187 / .082 / .194	.151 / -.127 / -.041	.168 / .271 / .101	.430 / .291 / .436	**.197** / .158 / **.169**
		11	.148 / .338 / **.374**	- / .132 / .266	.465 / .275 / **.435**	-.057 / **.175** / **.133**	.351 / .310 / .039	-.004 / .034 / -.069	.068 / .141 / **.279**	.157 / .113 / **.262**	.021 / -.232 / -.211	.090 / .146 / .062	.322 / .223 / .243	.151 / .151 / .158
		12	.289 / .181 / .278	- / **.277** / .398	**.469** / .280 / .224	-.090 / .023 / -.076	.225 / .067 / .224	.069 / .017 / -.068	**.279** / .086 / .209	.094 / -.116 / .130	**.314** / **.035** / -.100	.011 / -.090 / .030	.165 / .465 / **.448**	.179 / .077 / .142
	WiDiD	1	.253 / .301 / .278	- / .028 / -.048	.147 / .204 / .219	.120 / .052 / -.062	.132 / .051 / -.015	.159 / .047 / .125	.108 / .073 / .197	.090 / -.036 / .051	.356 / .150 / .090	.120 / .127 / .154	.122 / .026 / .160	.146 / .074 / .103
		2	.434 / .261 / .065	- / .018 / -.130	.106 / .143 / .292	-.041 / .015 / -.118	.103 / .105 / .110	.209 / -.046 / .274	.076 / .180 / .060	.212 / -.038 / -.008	.285 / -.030 / .085	.161 / .103 / .214	.371 / -.013 / .063	.175 / .060 / .094
		3	.423 / .268 / .147	- / .026 / .019	.115 / .120 / .474	.198 / .029 / .106	.228 / .108 / .118	**.251** / -.073 / **.345**	.091 / .113 / .184	.233 / .077 / .153	.229 / -.102 / .074	.239 / .064 / .204	.256 / .114 / .349	.216 / .065 / .203
		4	**.611** / .228 / .448	- / .030 / .108	.126 / .067 / .424	.176 / -.130 / .312	.292 / .175 / .221	.091 / -.039 / .332	.010 / .041 / .307	.157 / -.053 / .059	.242 / .038 / .002	.340 / .152 / .062	.388 / .279 / **.417**	.200 / .054 / .244
		5	.527 / .078 / .393	- / -.020 / -.037	.190 / .173 / **.509**	.151 / -.074 / .300	.356 / .295 / .310	-.034 / .023 / .259	.071 / .076 / .314	.205 / .137 / .202	**.297** / .100 / .023	.380 / .156 / .316	.524 / .193 / .217	.218 / .112 / .265
		6	.458 / .250 / .625	- / -.030 / -.050	.293 / .294 / .433	.211 / .148 / .335	.382 / .387 / .346	.094 / .063 / .184	.141 / .066 / .210	.182 / **.288** / .264	.261 / -.080 / .215	.428 / .295 / .102	.446 / .271 / .335	.252 / .185 / .269
		7	**.305** / .328 / .475	- / **.139** / .106	.235 / .253 / .514	**.295** / .198 / **.414**	.382 / .318 / .324	.017 / .032 / .292	.203 / **.285** / .152	.216 / .188 / **.458**	.244 / .119 / .247	.397 / .195 / -.034	.338 / .298 / .293	.237 / .211 / .304
		8	.449 / .312 / .411	- / .091 / .038	.344 / .341 / .565	.071 / **.354** / .321	.340 / .371 / .395	.000 / -.008 / .105	.284 / .260 / .243	.025 / .203 / .267	.221 / .226 / .262	.449 / **.428** / .155	.475 / .325 / .286	.224 / **.242** / .271
		9	.544 / **.509** / .567	- / -.066 / .104	.353 / .299 / .573	.184 / .319 / .203	.324 / **.450** / .372	-.002 / .075 / .108	.083 / .076 / .171	.205 / .205 / .388	.183 / .063 / .174	.390 / .118 / .149	.404 / .347 / .328	.222 / .212 / .280
		10	.396 / .301 / .587	- / -.024 / **.187**	.315 / **.407** / .477	.145 / .233 / .148	.306 / .388 / **.471**	.011 / .087 / .270	**.302** / .090 / .308	.060 / .172 / .328	.155 / .179 / .234	**.488** / .175 / .275	.428 / **.355** / .383	.224 / .204 / .339
		11	.299 / .218 / **.627**	- / -.064 / -.111	.258 / .381 / .486	.172 / .128 / .343	**.424** / .432 / .464	.134 / **.152** / .220	.234 / .120 / **.334**	.185 / .087 / .312	.218 / .195 / **.345**	.296 / .291 / **.438**	**.539** / .277 / .372	**.260** / .199 / **.345**
		12	.385 / .323 / .564	- / -.039 / -.064	**.355** / .312 / .499	.106 / .195 / .129	.383 / .343 / .459	.135 / -.068 / .268	.102 / .160 / .216	**.243** / .142 / .342	.233 / **.241** / .226	.087 / .290 / .349	.533 / .338 / .382	.239 / .181 / .314

Table B.1: Comprehensive evaluation of standard approaches to GCD by using the layers 1-12 of **BERT / mBERT / XLM-R**. Top score for each approach, model, and benchmark in **bold**. Avg is the weighted average score based on the number of targets in each benchmark.

B.2 Optimal layer combinations for Graded Change Detection

For the sake of comparison, we report in Table B.2 the overall top score for GCD obtained using BERT, mBERT, and XLM-R. Specifically, we present results for the optimal combination and the outcome obtained by summing the last four layers, separated by a slash. Additionally, we include the standard result obtained using the last layer individually.

]

			EN	**LA**	**DE**	**SV**	**ES**		**RU**			**NO**	**ZH**
			$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_1 - C_2$	$C_2 - C_3$	$C_1 - C_3$	$C_1 - C_2$	$C_2 - C_3$	$C_1 - C_2$
form-based	APD	BERT	**.692** / .566 *(.563)*	/	.412 / .349 *(.271)*	.325 / .272 *(.270)*	**.488** / .310 *(.335)*	**.573** / .537 *(.518)*	**.506** / .477 *(.482)*	**.546** / .522 *(.476)*	**.463** / .457 *(.441)*	**.556** / .521 *(.466)*	**.760** / .658 *(.656)*
		mBERT	.466 / .365 *(.363)*	.136 / .034 *(.102)*	.468 / .370 *(.398)*	.486 / .398 *(.389)*	.423 / .351 *(.341)*	.419 / .365 *(.368)*	.393 / .324 *(.345)*	.443 / .386 *(.386)*	.320 / .248 *(.279)*	.496 / .429 *(.488)*	.739 / .674 *(.689)*
		XLM-R	.579 / .518 *(.444)*	.080 / -.072 ***(.151)***	**.496** / .438 *(.264)*	**.496** / **.496** *(.257)*	.443 / .398 *(.386)*	.441 / .368 *(.290)*	.491 / .404 *(.287)*	.432 / .397 *(.318)*	.215 / .180 *(.195)*	.421 / .418 *(.379)*	.675 / .627 *(.500)*
	PRT	BERT	**.550** / .520 *(.457)*	/	.421 / .397 *(.422)*	**.293** / .170 *(.158)*	.478 / .441 *(.413)*	**.425** / .368 *(.400)*	.418 / .374 *(.374)*	.383 / .346 *(.347)*	**.538** / .513 *(.507)*	**.513** / .481 *(.444)*	.706 / .649 *(**.712**)*
		mBERT	.382 / .339 *(.270)*	.352 / .305 *(.380)*	.467 / .454 *(.436)*	.132 / .105 *(.193)*	**.555** / .514 *(.543)*	.411 / .373 *(.391)*	.442 / .386 *(.356)*	.434 / .367 *(.423)*	.256 / .228 *(.219)*	.432 / .405 *(.438)*	.648 / .588 *(.524)*
		XLM-R	.513 / .476 *(.411)*	.365 / .312 ***(.424)***	**.497** / .486 *(.369)*	.253 / .236 *(.020)*	.538 / .522 *(.505)*	.409 / .402 *(.320)*	**.530** / .453 *(.443)*	**.449** / .435 *(.405)*	.384 / .384 *(.387)*	.270 / .220 *(.149)*	.642 / .627 *(.558)*
sense-based	AP	BERT	.464 / .245 *(.289)*	/	**.520** / .435 *(.469)*	.201 / -.061 *(-.090)*	**.499** / .295 *(.225)*	.292 / .149 *(.069)*	**.418** / .216 *(.279)*	**.386** / .207 *(.094)*	**.329** / .028 *(.314)*	.466 / .227 *(.011)*	**.671** / .587 *(.165)*
		mBERT	**.501** / .313 *(.181)*	.326 / .179 *(.277)*	.428 / .329 *(.280)*	.193 / .090 *(.023)*	.484 / .259 *(.067)*	**.209** / .123 *(.017)*	.316 / .175 *(.086)*	.247 / .058 *(-.116)*	.194 / -.105 *(.035)*	**.539** / .275 *(-.090)*	.645 / .256 *(465)*
		XLM-R	.473 / .340 *(.278)*	**.482** / .398 *(.398)*	.502 / .370 *(.224)*	**.235** / .022 *(-.076)*	.307 / .170 *(.224)*	.162 / .012 *(-.068)*	.378 / .247 *(.209)*	.358 / .224 *(.130)*	.322 / .132 *(-.100)*	.465 / .035 *(.030)*	.583 / .135 *(.448)*
	WiDiD	BERT	.635 / .441 *(.385)*	/	.465 / .322 *(.355)*	.432 / .177 *(.106)*	.466 / .361 *(.383)*	.388 / .136 *(.135)*	.410 / .190 *(.102)*	.408 / .280 *(.243)*	**.531** / .160 *(.233)*	**.578** / .336 *(.087)*	**.701** / .537 *(.533)*
		mBERT	.600 / .317 *(.323)*	.252 / .055 *(-.039)*	.610 / .422 *(.312)*	**.521** / .413 *(.195)*	**.575** / .272 *(.343)*	.255 / .215 *(-.068)*	.373 / .056 *(.160)*	.327 / .252 *(.142)*	.500 / .459 *(.241)*	.467 / .292 *(.290)*	.620 / .513 *(.338)*
		XLM-R	**.760** / .663 *(.564)*	**.347** / -.077 *(-.064)*	**.721** / .557 *(.499)*	.503 / .220 *(.129)*	.526 / .437 *(.459)*	**.426** / .223 *(.268)*	**.460** / .352 *(.216)*	**.485** / .304 *(.342)*	.505 / .399 *(.226)*	.440 / .336 *(.349)*	.637 / .349 *(.382)*

Table B.2: Top score for GCD obtained using BERT, mBERT, and XLM-R. We present results for the optimal combination and the outcome obtained by summing the last four layers, separated by a slash (i.e., best results **/** sum of last four layers). Additionally, for comparison purposes, we include the result obtained using the last layer individually *(enclosed in brackets).* Top scores for approach and benchmark are highlighted in **bold**.

Chapter C

Analyzing Semantic Change through lexical replacements

This appendix contains material for Chapter 8.

C.0.1 Artificial diachronic corpus

We generated an artificial diachronic corpus for LSC by utilising the SemEval and LSCDiscovery benchmakrs for LSC in DWUG format[1] (see Table C.1). Instead of incorporating data from both time periods, T_1 and T_2, we discarded information from the first time period as it is more likely to contain word meanings outside the pre-trained knowledge of the models under examination. We created two distinct artificial subcorpora, C_1 and C_2, by randomly sampling occurrences from the data of the second time period T_2. The DWUG English dataset contains data for 46 target words.

For each target t, we considered all sentences where another target $t1$, with $t1 \neq t$, appeared as potential candidates to emulate instances of semantic change. We simulated a change instance through a *random* replacement, that is by replacing t in the sentence where $t1$ occurred – i.e., $t1 \leftarrow t$. We sample a varying number of sentences and perform replacements for each target, thereby emulating a varying degree of semantic change.

[1] English: https://zenodo.org/records/5796878, German: https://zenodo.org/records/5796871, Swedish: https://zenodo.org/records/5090648, Spanish: https://zenodo.org/records/6433667

References	Benchmark	# targets
Schlechtweg et al., 2020	DWUG-English	46
Schlechtweg et al., 2020	DWUG-German	50
Schlechtweg et al., 2020	DWUG-Swedish	44
Zamora-Reina et al., 2022b	DWUG-Spanish	100

Table C.1: References and number of targets for each consider artificial corpus.

Chapter D

Automatically generated definitions and their utility for modeling word meaning

This appendix contains material for Chapter 9.

D.1 Fine-tuning

In our experiments, we conducted multiple rounds of fine-tuning, systematically testing various parameters. Specifically, we detail these configurations in Table D.1. In line with Huerta-Enochian (2024), who recently demonstrated that prompt loss can be safely ignored for many datasets, we observed lower preliminary results in the evaluation tasks for models chosen based on validation performance. Therefore, we selected the final models (see Table D.2) based on the checkpoint from the last training epoch that had the best performance on the Definition Generation task.

D.1.1 Lora rank-alpha

We conduct fine-tuning using LoRA, (Hu et al., 2021) and QLORA, (Dettmers et al., 2023) obtaining very similar evaluation results. Drawing from insights from prior research (Munoz et al., 2024) as well recent online discussions, we adopted a strategy where the LoRA alpha α was set to double the LoRA rank r. In our experiments for the Definition Generation task, larger ranks resulted in higher performance on **WordNet** and slightly higher performance on **Oxford** benchmarks. However, no improvement was noted for **Wiktionary** (see Figure D.1).

D.2 SBERT models

In our experiments, we made an effort to evaluate all the Bi-Encoder SBERT models available at `https://sbert.net/` (see Table D.3). This thorough assessment ensures that our findings are robust and accurate.

Parameter	Experimented values
Model	*Meta-Llama-3-8B-Instruct*, *Llama-2-7b-chat-hf*
GPU	A100:fat (80 GB)
Hours	7-8
PEFT	LoRA, QLoRA
Dropout	0.05, 0.1, 0.2
Weight decay	0.001, 0.0001
Learning rate	1e-4, 1e-5
Lora ranks	8, 32, 64, 128, 256, 512, 1024
Lora alpha	16, 64, 256, 512, 1024, 2048
Warmup ratio	0.03, 0.05
Eval steps	250
Train epochs	4, 5, 10
Max seq. length	512
Batch size	32
Optimizer	Adam
LoRA target modules	q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj, lm_head

Table D.1: Settings and parameters used during training. Parameters shown in small font represent preliminary experiments that were not further evaluated.

Final setting	Llama2Dictionary	Llama3Dictionary
GPU	A100:fat (80 GB)	A100:fat (80 GB)
Hours	7-8	8-9
PEFT	LoRA	LoRA
Dropout	0.1	0.05
Weight decay	0.001	0.001
Learning rate	1e-4	1e-4
Lora ranks	1024	512
Lora alpha	2048	1024
Warmup ratio	0.05	0.05
Eval steps	epochs	epochs
Train epochs	4	4
Max seq. length	512	512
Batch size	32	32
Optimizer	Adam	Adam
LoRA target modules	q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj, lm_head	q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj, lm_head

Table D.2: Parameters of our final models. Our code is publicly available at `https://github.com/FrancescoPeriti/LlamaDictionary` for further details. For finetuning, we rely on the transformers library (Wolf et al., 2020).

While we acknowledge that other models may exist, the evaluation results we present remain valuable and consistent across the models tested, contributing to the broader perspective presented in the paper.

Further parameters are related to our procedure for addressing the Word-in-Context, Word Sense Induction, and Lexical Semantic Change tasks. We report these parameters in Table D.5.

SBERT models
all-mpnet-base-v2
multi-qa-mpnet-base-dot-v1
all-distilroberta-v1
all-MiniLM-L12-v2
multi-qa-distilbert-cos-v1
all-MiniLM-L6-v2
multi-qa-MiniLM-L6-cos-v1
paraphrase-multilingual-mpnet-base-v2
paraphrase-albert-small-v2
paraphrase-multilingual-MiniLM-L12-v2
paraphrase-MiniLM-L3-v2
distiluse-base-multilingual-cased-v1
distiluse-base-multilingual-cased-v2

Table D.3: Experimented SBERT models. We report in **bold** the model used for the results obtained in the main paper. We use this model as it was used in previous experiments by Giulianelli et al. (2023).

D.3 Definition Generation

In our work, we extensively evaluated our `LlamaDictionary` models along with the `Flan-T5-Definition` models by Giulianelli et al. (2023), setting new state-of-the-art results on the Definition Generation tasks across multiple benchmarks. In Table D.6, we provide a full comparison, including individual scores for each benchmark and the measures considered.

Benchmark	**Target** w	**Example** *e*	**Definition** *e*
WordNet	accuracy	He was beginning to doubt the *accuracy* of his compass	The quality of being near to the true value
Oxford	accuracy	However, these studies have not generally had enough participants to provide precise estimates of *accuracy*.	The quality or state of being correct or precise
Wiktionary	accuracy	The efficiency of the instrument will also depend upon the *accuracy* with which the piston fits the bottom and sides of the barrel. When the piston is depressed to the bottom, it is considered in theory to be in absolute contact, so as to exclude every particle of air from the space between it and the bottom.	The state of being accurate; being free from mistakes, this exemption arising from carefulness; exactness; correctness
Oxford	yesterday	*Yesterday* the weather was beautiful	On the day preceding today
Oxford	yesterday	It was in *yesterday* 's newspapers	The day immediately before today
Oxford	yesterday	I am doing a research paper on women 's voting rights ; *yesterday* and today	On the day before today
Oxford	yesterday	On a day like today after *yesterday* , i tend to reflect , internalize , and re-address the balance	The day before today

Table D.4: Example of correct but inconsistent definitions from the considered benchmarks. It is unnecessary to train the model to provide different answers. Ideally, a single definition should be used for different examples of the considered target.

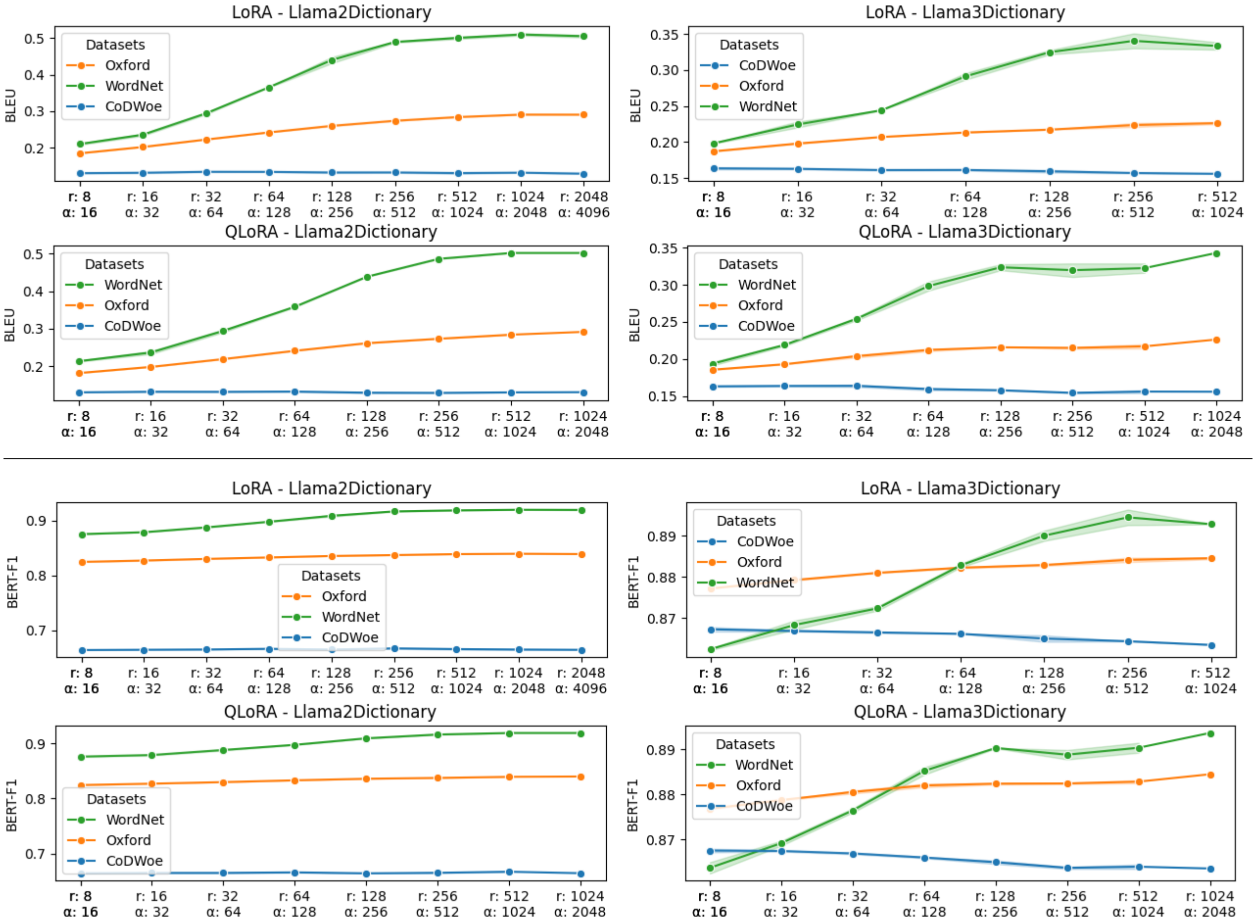


Figure D.1: Average performance of trained models using LoRA (Hu et al., 2021) and QLoRA (Dettmers et al., 2023) with parameters from Table D.1. We conducted experiments with LoRA *alpha* α set to double the *rank* r and observed that larger ranks resulted in higher performance on **WordNet** and slightly higher performance on **Oxford** benchmarks. However, no improvement was noted for **Wiktionary**. We report BERT-F1 and BLEU as examples. Similar trends were observed for other performance metrics.

	Evaluation tasks			
	DG	**WiC**	**WSI**	**LSC**

gen. model	`LlamaDictionary,`
	`Flan-T5-Definition`
		`LlamaDictionary,`
		`Flan-T5-Definition`
			`LlamaDictionary,`
			`Flan-T5-Definition`
				`LlamaDictionary,`
				`Flan-T5-Definition`

temperature	0.0	0.0	0.0	0.0
enc. model	`roberta-large`	`all-distilroberta-v1`	`all-distilroberta-v1`	`all-distilroberta-v1`
metric	BERTScore	cosine	cosine	cosine (APD) canberra (APDP) following Periti et al.; Periti and Tahmasebi
clustering	-	-	HDBSCAN	HDBSCAN
HDBSCAN-allow_single_cluster	-	-	True	True
HDBSCAN-min_cluster_size	-	-	2	2
HDBSCAN-cluster_selection_method	-	-	leaf	leaf

Table D.5: Models and parameters used for addressing the DG, WIC, WSI, and LSC tasks. We rely on the HDBSCAN implementation of the scikit-learn library (Pedregosa et al., 2011).

	ROUGE-L	BLEU	BERT-F1	NIST	SACREBLEU	METEOR	EXACT MATCH
WordNet - seen							
Noraset et al. (2017)	-	.236*	-	.497*	-	-	-
Ni and Wang (2017)	-	.248*	-	.403*	-	-	-
Gadetsky et al. (2018)	-	.237*	-	.443*	-	-	-
Ishiwatari et al. (2019)	-	.248	-	.435*	-	-	-
Huang et al. (2021)	-	.327	-	.646	-	-	-
Zhang et al. (2022)	-	.320	-	.747	-	-	-
Giulianelli et al. (2023) Reported	.522	.328	**.921**	-	-	-	-
Giulianelli et al. (2023) Observed	.405	.320	.893	.907	23.302	.374	.164
Llama2chat	**.564**	**.513**	**.920**	**1.391**	**41.096**	**.536**	**.373**
Llama3Instruct	.435	.339	.893	1.012	27.400	.480	.131
Oxford - seen							
Noraset et al. (2017)	-	.149*	-	.327*	-	-	-
Ni and Wang (2017)	-	.176*	-	.313*	-	-	-
Gadetsky et al. (2018)	-	.120	-	.358*	-	-	-
Ishiwatari et al. (2019)	-	.185	-	.382*	-	-	-
Huang et al. (2021)	-	.265	-	.742	-	-	-
Bevilacqua et al. (2020)	.294	.088	.768	-	-	.135	-
Zhang et al. (2022)	-	.271	-	.794	-	-	-
Giulianelli et al. (2023) Reported	.387	.186	**.897**	-	-	-	-
Giulianelli et al. (2023) Observed	.324	.213	.878	.749	14.400	.292	.057
Llama2chat	**.398**	**.291**	.840	**.969**	**21.410**	.367	**.158**
Llama3Instruct	.365	.228	**.885**	.900	16.550	**.373**	.055
Wikitionary - seen							
Llama2chat	.222	.131	.666	.408	6.963	.183	.025
Llama3Instruct	**.267**	.156	**.863**	.517	**8.100**	**.232**	**.034**
Urban - unseen							
Noraset et al. (2017) - seen	-	.515*	-	.104*	-	-	-
Ni and Wang (2017) - seen	-	**.899***	-	.174*	-	-	-
Gadetsky et al. (2018) - seen	-	.088*	-	.194*	-	-	-
Ishiwatari et al. (2019) - seen	-	.105	-	.192*	-	-	-
Huang et al. (2021) - seen	-	.177	-	.355	-	-	-
Zhang et al. (2022) - seen	-	.194	-	**.410**	-	-	-
Giulianelli et al. (2023) - unseen Observed	.106	.053	.835	.167	2.160	.068	**.001**
Llama2chat - unseen	.110	.055	.812	.170	**2.247**	.071	**.001**
Llama3instruct - unseen	**.115**	.057	**.836**	.197	2.331	**.079**	**.001**
Wikipedia - unseen							
Noraset et al. (2017) - seen	-	.446*	-	.334*	-	-	-
Ni and Wang (2017) - seen	-	.527*	-	.552*	-	-	-
Gadetsky et al. (2018)- seen	-	.450*	-	.331*	-	-	-
Ishiwatari et al. (2019)- seen	-	.538	-	.567*	-	-	-
Huang et al. (2021)- seen	-	**.556**	-	**.640**	-	-	-
Giulianelli et al. (2023) - unseen Observed	.240	.138	**.863**	.511	8.212	.263	**.000**
Llama2chat - unseen	.213	.123	.716	.523	7.399	.232	**.000**
Llama3Instruct - unseen	**.253**	**.144**	**.863**	.614	**8.638**	**.290**	**.000**

Table D.6: Evaluation results for the **Definition Generation** task. The best result is highlighted in bold. Our model is trained exclusively on the training set of the WordNet, Oxford, and Wiktionary datasets. Results marked with * are reported from experiments in Huang et al. (2021).

Chapter E

Modeling historical resonance

This appendix contains material for Chapter 10.

E.1 Train-Dev-Test partitions

For each randomized split, we use the filtered instances (see Section 10.4.2) to create the Train-Dev-Test partitions, comprising approximately 80%, 10%, and 10% of the instances, respectively. In the creation of the Train set of a split, we exclude the $\langle t, c_1, c_2 \rangle$ instances associated to four targets t (i.e., 10% of the benchmark's targets). We include these instances in Dev and Test to enforce the Out-of-Vocabulary (OOV) evaluation. Specifically, we include in Dev the instances associated with two targets, and in Test the instances of the remaining excluded targets.

Notably, we ensure that each partition has a distinct set of OOV targets, such that the intersection of the OOV sets for each split is empty.

E.2 Model evaluation

We evaluate almost all the pre-trained models available at `https://www.sbert.net/index.html`. Specifically, we considered only pre-trained models trained on tasks based on textual similarity and excluded those trained on other tasks (e.g., models for Image Search). Table E.2 reports results for all the evaluated models.

For the sake of transparency and completeness, we have included the computation of Precision (PR) and Recall (RE) for each considered class. Specifically, for label 1, PR and RE are calculated as $\frac{TP}{(TP+FP)}$ and $\frac{TP}{(TP+FN)}$ respectively. Similarly, for label 0, PR and RE are computed as $\frac{TN}{(TN+FN)}$ and $\frac{TN}{(TN+FP)}$. In scientific literature, these latter metrics are also known as Negative Predictive Value and Sensitivity. For the sake of clarity, we preferred using PR and RE for *label 0* and *label 1* instead of distinguishing between Precision (PR), Recall (RE), Negative Predictive Value (NPV), and Specificity (SP).

E.3 Fine-tuning

For each randomized split, we fine-tuned each considered model on the Train set and subsequently validated its performance on the Dev set. To do this, we employed the AdamW optimizer, coupled with a linear learning rate warm-up applied to the first 10% of the Train set. We used grid search to optimize hyper-parameters, with a particular focus on fine-tuning the learning rate by testing values from the set {1e-6, 2e-6, 5e-6, 1e-5, 2e-5}. We do not use weight decay, since our initial experiments did not yield any additional benefits. During the training, we leveraged an early stopping strategy. In particular, we fine-tuned each pre-trained model on TRiC instances using the *contrastive loss* (Hadsell et al., 2006). This loss minimizes the distance between embeddings of similar sentences and maximizes the distance for dissimilar sentences. We finally ceased training when there was no further improvement observed on the Dev set. Details on the setup of hyper-parameters are shown in Table E.1.

E.4 Hyper-parameters

Models	Learning Rate
all-distilroberta-v1 (ADR)	1e-05
distiluse-base-multilingual-cased-v1 (DBM)	1e-05
paraphrase-multilingual-MiniLM-L12-v2 (PAM)	2e-05
paraphrase-multilingual-mpnet-base-v2 (PAR)	5e-06
multi-qa-mpnet-base-cos-v1 (MQA)	1e-05

Table E.1: Models learning rates.

E.5 Annotation

Annotating topic relatedness, instead of relying on explicit topic labels, closely resembles recent work exemplified in the Word-in-Context task (Pilehvar and Camacho-Collados, 2019), which relies on annotating word meaning relatedness rather than explicit sense labels. The methodology underlying this approach is thoroughly elucidated in our guidelines, submitted as supplementary material along with our paper. The topic relatedness is evaluated by using the four-point DURel relatedness scale (see Figure 10.1). Annotator were trained in a 30-minute online session and tested on a small set of 25 instances (tutorial). In particular, we ensured that each annotator achieved a minimum agreement (measured by Spearman correlation) of at least .550 with the tutorial judgments. We interpreted these results as reliable, and consequently, we proceeded with the annotation of our benchmark. Then, we derive TRiC and TRaC labels after conducting an empirical analysis of the agreement of each level of our topic relatedness scale (see Section 10.4.2).

	Standard Test Set								**Out-of-vocabulary (OOV) Test set**							
	Label 0			*Label 1*			***All***		*Label 0*			*Label 1*			***All***	
Models	PR	RE	F1	PR	RE	F1	F1	SP	PR	RE	F1	PR	RE	F1	F1	SP
paraphrase-multilingual-MiniLM-L12-v2 **(PAM)**	**.96±.02**	.46±.09	.61±.08	.41±.09	.96±.02	.57±.08	.61±.07	**.58±.08**	.96±.04	.43±.17	.57±.16	.37±.15	.95±.05	.52±.15	.59±.12	**.49±.22**
+*MASK*	*.89±.05*	*.88±.06*	*.88±.03*	*.71±.10*	*.72±.10*	*.70±.05*	*.83±.03*	*.67±.04*	*.89±.09*	*.86±.09*	*.87±.06*	*.65±.19*	*.71±.18*	*.65±.12*	*.83±.05*	*.60±.13*
multi-qa-mpnet-base-cos-v1 **(MQA)**	.94±.03	.42±.11	.58±.11	.40±.10	.94±.03	.55±.09	.58±.09	.55±.09	.94±.09	.39±.19	.53±.20	.36±.19	.96±.03	.50±.18	.55±.16	**.49±.21**
+*MASK*	*.88±.05*	*.87±.07*	*.88±.04*	*.71±.10*	*.71±.12*	*.69±.06*	*.83±.04*	*.68±.05*	*.89±.07*	*.86±.10*	*.87±.06*	*.63±.18*	*.69±.16*	*.63±.13*	*.83±.05*	*.62±.13*
all-distilroberta-v1 **(ADR)**	.95±.03	.47±.13	.62±.11	.42±.11	.93±.04	.57±.10	.61±.10	.55±.09	.94±.07	.45±.20	.58±.20	.38±.19	.93±.06	.51±.18	.58±.16	.48±.20
+*MASK*	*.89±.05*	*.87±.07*	*.87±.03*	*.70±.14*	*.72±.12*	*.69±.07*	*.82±.03*	*.67±.06*	*.90±.07*	*.85±.10*	*.87±.05*	*.62±.21*	*.71±.18*	*.63±.14*	*.82±.05*	*.62±.15*
all-mpnet-base-v2	.93±.03	.48±.14	.62±.13	.42±.12	.91±.03	.57±.10	.61±.11	.53±.10	.93±.09	.44±.22	.56±.21	.38±.20	.94±.05	.51±.18	.57±.18	.48±.20
+*MASK*	*.88±.06*	*.84±.09*	*.85±.05*	*.66±.12*	*.71±.11*	*.67±.04*	*.81±.04*	*.66±.06*	*.89±.08*	*.82±.11*	*.85±.07*	*.59±.20*	*.73±.14*	*.62±.11*	*.81±.05*	.61±.15
paraphrase-multilingual-mpnet-base-v2 **(PAR)**	.95±.03	.40±.10	.56±.09	.39±.09	.95±.04	.55±.08	.56±.07	.56±.09	.93±.11	.35±.18	.49±.19	.34±.15	.95±.06	.49±.16	.52±.15	.47±.25
+*MASK*	*.89±.05*	*.85±.07*	*.87±.04*	*.69±.10*	*.75±.11*	*.70±.05*	*.83±.03*	*.68±.03*	*.90±.08*	*.83±.13*	*.86±.07*	*.63±.19*	*.75±.17*	*.65±.10*	*.82±.05*	.62±.11
all-MiniLM-L12-v2	.95±.03	.40±.12	.55±.11	.39±.11	.94±.04	.54±.10	.55±.10	.52±.08	.94±.03	.37±.18	.50±.18	.35±.18	.93±.06	.48±.18	.52±.16	.47±.17
+*MASK*	*.88±.05*	*.87±.08*	*.87±.03*	*.70±.13*	*.72±.11*	*.69±.06*	*.82±.03*	*.68±.04*	*.89±.08*	*.85±.10*	*.86±.05*	.62±.21	*.71±.17*	*.62±.14*	*.82±.04*	*.62±.13*
multi-qa-distilbert-cos-v1	**.96±.03**	.33±.11	.48±.11	.37±.10	**.97±.02**	.53±.09	.50±.10	.53±.09	**.97±.06**	.29±.18	.42±.19	.33±.16	**.97±.05**	.47±.17	.46±.16	.47±.21
+*MASK*	*.88±.06*	*.86±.06*	*.87±.03*	*.68±.11*	*.73±.10*	*.69±.05*	*.82±.03*	*.68±.05*	*.89±.10*	*.85±.08*	*.86±.05*	*.61±.19*	*.71±.14*	*.63±.11*	*.82±.05*	*.62±.14*
multi-qa-mpnet-base-dot-v1	.92±.05	.48±.14	.62±.11	.42±.11	.89±.06	.56±.09	.61±.09	.51±.10	.91±.15	.45±.19	.59±.17	.38±.18	.92±.07	.51±.17	.59±.14	.46±.22
+*MASK*	*.87±.07*	*.86±.08*	*.86±.03*	*.69±.12*	*.65±.19*	*.63±.08*	*.80±.03*	*.63±.05*	*.87±.09*	.85±.09	*.85±.06*	*.62±.23*	*.63±.20*	*.57±.13*	*.80±.05*	*.57±.12*
all-MiniLM-L6-v2	**.96±.02**	.40±.10	.55±.10	.39±.10	.95±.04	.55±.09	.56±.08	.53±.09	**.97±.03**	.37±.17	.51±.19	.35±.17	.95±.07	.49±.18	.54±.14	.44±.23
+*MASK*	*.88±.05*	*.88±.06*	*.88±.03*	*.72±.12*	*.70±.12*	*.69±.06*	*.83±.03*	*.67±.05*	*.89±.07*	*.88±.09*	*.88±.05*	*.67±.22*	*.66±.19*	*.62±.14*	*.83±.04*	*.61±.16*
distiluse-base-multilingual-cased-v1 **(DBM)**	**.96±.02**	.26±.12	.40±.14	.35±.09	**.97±.03**	.51±.09	.43±.12	.54±.09	.96±.08	.21±.19	.31±.23	.31±.14	**.97±.05**	.45±.16	.38±.18	.44±.23
+*MASK*	*.87±.07*	*.88±.07*	*.87±.03*	*.72±.14*	*.66±.16*	*.66±.09*	*.81±.03*	*.64±.04*	*.88±.09*	*.88±.09*	*.87±.05*	*.66±.23*	*.64±.25*	*.58±.19*	*.82±.04*	*.58±.12*
distiluse-base-multilingual-cased-v2	**.96±.03**	.26±.08	.40±.10	.34±.09	.97±.03	.50±.09	.43±.09	.54±.10	.96±.08	.21±.16	.32±.20	.30±.15	.96±.08	.44±.17	.38±.16	.44±.25
+*MASK*	*.87±.06*	*.89±.07*	*.87±.03*	*.72±.14*	*.66±.14*	*.66±.09*	*.82±.03*	*.65±.04*	*.88±.08*	*.88±.10*	*.87±.05*	*.66±.24*	*.64±.23*	*.60±.18*	*.82±.05*	*.59±.12*
multi-qa-distilbert-dot-v1	.93±.04	.40±.12	.55±.11	.39±.09	.92±.05	.54±.09	.56±.09	.51±.09	.92±.12	.36±.16	.50±.16	.34±.15	.92±.07	.48±.16	.53±.11	.43±.19
+*MASK*	*.85±.05*	*.87±.08*	*.85±.03*	*.69±.15*	*.60±.16*	*.61±.08*	*.79±.02*	*.62±.05*	*.86±.09*	*.87±.09*	*.86±.05*	*.66±.24*	*.58±.22*	*.55±.16*	*.80±.03*	*.57±.14*
paraphrase-albert-small-v2	**.96±.02**	.36±.09	.52±.09	.38±.09	.96±.02	.54±.09	.53±.07	.53±.09	.95±.10	.32±.16	.46±.18	.33±.14	**.97±.04**	.48±.16	.50±.12	.43±.25
+*MASK*	*.88±.06*	*.84±.07*	*.86±.03*	*.65±.11*	*.70±.14*	*.66±.07*	*.80±.02*	*.65±.05*	*.88±.08*	*.82±.12*	*.84±.07*	*.56±.19*	*.67±.20*	*.58±.14*	*.80±.05*	*.57±.14*
multi-qa-MiniLM-L6-cos-v1	.95±.03	.37±.09	.52±.09	.38±.10	.95±.04	.53±.10	.53±.08	.52±.10	.91±.14	.34±.18	.48±.19	.34±.17	.94±.08	.47±.17	.50±.16	.42±.25
+*MASK*	*.88±.05*	*.88±.04*	*.88±.02*	*.70±.09*	*.69±.09*	*.68±.06*	*.83±.02*	*.66±.04*	*.87±.09*	*.87±.07*	*.87±.05*	*.61±.19*	*.64±.19*	*.60±.14*	*.82±.04*	*.60±.15*
stsb-roberta-large	.33±.08	**.99±.02**	.49±.09	**.97±.03**	.19±.10	.30±.13	.36±.11	.52±.07	.29±.14	**.99±.03**	.42±.16	**.94±.15**	.11±.15	.18±.19	.28±.15	.42±.20
+*MASK*	*.70±.13*	*.68±.15*	*.66±.07*	*.87±.06*	*.87±.08*	*.87±.03*	*.81±.03*	*.66±.04*	*.62±.27*	*.64±.28*	*.57±.21*	*.87±.10*	*.86±.11*	*.86±.06*	*.80±.06*	*.62±.08*
paraphrase-MiniLM-L3-v2	.95±.03	.28±.09	.43±.10	.35±.08	.96±.03	.51±.09	.46±.08	.49±.11	.96±.05	.23±.19	.34±.21	.31±.14	**.97±.05**	.45±.16	.41±.17	.40±.27
+*MASK*	*.87±.05*	*.86±.09*	*.86±.04*	*.68±.12*	*.68±.11*	*.66±.05*	*.81±.03*	*.65±.04*	*.88±.07*	*.85±.12*	*.86±.06*	*.61±.21*	*.64±.22*	*.59±.15*	*.81±.05*	*.59±.14*
msmarco-distilbert-dot-v5	.93±.04	.36±.10	.51±.09	.37±.09	.93±.03	.52±.08	.52±.08	.47±.08	.92±.09	.31±.18	.43±.20	.32±.14	.92±.08	.46±.15	.48±.13	.38±.19
+*MASK*	*.87±.05*	*.91±.04*	*.89±.03*	*.75±.10*	*.66±.08*	*.69±.06*	*.84±.03*	*.64±.04*	*.87±.09*	*.90±.05*	*.88±.06*	*.67±.18*	*.60±.16*	*.61±.15*	*.83±.06*	*.58±.10*
msmarco-MiniLM-L12-cos-v5	.91±.04	.44±.09	.59±.08	.39±.09	.90±.05	.54±.08	.58±.07	.44±.08	.91±.09	.44±.17	.58±.16	.36±.16	.88±.10	.49±.16	.59±.12	.38±.19
+*MASK*	*.85±.05*	*.88±.06*	*.86±.03*	*.68±.11*	*.60±.10*	*.62±.06*	*.80±.03*	*.59±.04*	*.85±.10*	*.88±.08*	*.86±.06*	.62±.21	*.55±.21*	*.53±.16*	*.79±.05*	*.54±.12*
multi-qa-MiniLM-L6-dot-v1	.89±.07	.54±.07	**.67±.06**	.42±.09	.84±.08	.55±.08	**.64±.05**	.46±.10	.87±.16	.51±.15	**.63±.15**	.37±.16	.83±.11	.49±.15	.62±.12	.37±.26
+*MASK*	*.83±.07*	*.86±.06*	*.84±.03*	*.61±.13*	*.56±.12*	*.56±.07*	*.76±.04*	*.53±.07*	*.82±.12*	*.86±.08*	*.83±.07*	*.53±.23*	*.50±.20*	*.47±.17*	*.76±.08*	*.45±.18*
msmarco-MiniLM-L6-cos-v5	.93±.03	.41±.10	.56±.10	.39±.09	.92±.06	.54±.09	.56±.08	.44±.10	.93±.07	.38±.18	.52±.18	.34±.16	.91±.12	.48±.17	.54±.14	.37±.22
+*MASK*	*.85±.06*	*.87±.07*	*.86±.04*	*.67±.10*	*.62±.14*	*.62±.07*	*.79±.03*	*.59±.04*	*.85±.11*	*.86±.09*	*.85±.07*	*.60±.17*	*.58±.24*	*.55±.16*	*.79±.05*	*.54±.11*
msmarco-distilbert-base-tas-b	.93±.04	.36±.13	.51±.13	.38±.09	.93±.05	.53±.09	.52±.11	.45±.10	.92±.10	.32±.22	.44±.21	.33±.15	.92±.10	.47±.16	.48±.17	.36±.23
+*MASK*	*.86±.07*	*.86±.08*	*.86±.03*	*.67±.14*	*.64±.14*	*.63±.07*	*.80±.03*	*.62±.05*	*.86±.11*	*.87±.11*	*.85±.06*	*.61±.23*	*.59±.26*	*.53±.20*	*.79±.07*	*.56±.14*
stsb-distilroberta-base	.33±.08	.96±.04	.49±.08	.94±.06	.23±.10	.35±.12	.40±.10	.43±.08	.29±.14	.96±.06	.43±.15	.89±.21	.17±.16	.27±.19	.34±.15	.36±.21
+*MASK*	*.66±.13*	*.61±.15*	*.61±.07*	*.85±.07*	*.86±.09*	*.85±.04*	*.78±.04*	*.59±.04*	*.58±.22*	*.56±.25*	*.51±.17*	*.85±.11*	*.84±.12*	*.84±.07*	*.77±.07*	*.55±.08*
msmarco-distilbert-cos-v5	.94±.03	.30±.09	.45±.11	.36±.09	.95±.03	.51±.09	.48±.09	.42±.09	.91±.12	.26±.14	.38±.17	.31±.14	.94±.06	.44±.15	.43±.13	.34±.17
+*MASK*	*.88±.05*	*.84±.06*	*.85±.03*	*.64±.10*	*.71±.11*	*.66±.07*	*.80±.02*	*.62±.03*	*.88±.08*	*.82±.08*	.84±.05	*.56±.19*	*.67±.16*	*.59±.15*	*.80±.04*	*.56±.09*
stsb-TinyBERT-L-4	.32±.09	.98±.03	.48±.10	.96±.03	.16±.13	.26±.16	.33±.14	.41±.07	.29±.14	.97±.05	.43±.17	.77±.39	.13±.18	.19±.23	.28±.20	.34±.19
+*MASK*	*.67±.15*	*.66±.16*	*.63±.07*	*.86±.07*	*.85±.09*	*.85±.04*	*.79±.04*	*.62±.04*	*.61±.23*	*.62±.26*	*.54±.17*	*.87±.10*	*.85±.11*	*.85±.05*	*.79±.05*	*.56±.11*
stsb-roberta-base	.31±.08	.98±.02	.47±.09	.95±.05	.13±.07	.22±.10	.30±.08	.42±.07	.28±.14	.97±.05	.41±.16	.91±.15	.10±.10	.16±.15	.26±.13	.33±.20
+*MASK*	*.68±.10*	*.64±.15*	*.64±.08*	*.86±.06*	*.87±.07*	*.86±.03*	*.80±.04*	*.63±.06*	*.57±.21*	*.57±.26*	*.52±.20*	*.86±.11*	*.86±.10*	*.85±.06*	*.78±.08*	*.57±.11*
msmarco-bert-base-dot-v5	.93±.03	.32±.10	.47±.11	.36±.08	.94±.03	.51±.09	.49±.09	.45±.09	.91±.07	.26±.19	.38±.21	.31±.14	.92±.08	.45±.16	.43±.15	.33±.24
+*MASK*	*.87±.05*	*.90±.05*	*.88±.03*	*.74±.11*	*.66±.09*	*.69±.06*	*.83±.03*	*.65±.03*	*.86±.09*	*.90±.06*	*.88±.05*	*.66±.18*	*.58±.20*	*.58±.17*	*.82±.05*	*.58±.11*
ms-marco-TinyBERT-L-2-v2	.32±.08	.97±.02	.48±.09	.93±.06	.17±.11	.28±.14	.34±.12	.34±.10	.29±.14	.97±.03	.43±.16	.78±.30	.13±.19	.20±.23	.29±.19	.26±.20
+*MASK*	*.67±.15*	*.64±.14*	*.63±.07*	*.86±.06*	*.86±.09*	*.85±.04*	*.79±.03*	*.60±.06*	*.60±.23*	*.61±.24*	*.55±.17*	*.87±.10*	*.86±.12*	*.85±.05*	*.79±.06*	*.55±.15*
ms-marco-MiniLM-L-2-v2	.32±.08	.97±.02	.48±.09	.94±.05	.16±.12	.26±.15	.33±.13	.36±.10	.29±.14	.97±.05	.43±.16	.91±.15	.13±.20	.19±.24	.29±.20	.26±.23
+*MASK*	*.67±.14*	*.61±.13*	*.62±.07*	*.85±.06*	*.87±.07*	*.85±.03*	*.79±.03*	*.57±.08*	*.58±.22*	*.55±.25*	*.50±.18*	*.85±.11*	*.85±.10*	*.84±.06*	*.77±.07*	*.51±.16*
ms-marco-MiniLM-L-4-v2	.32±.08	.95±.03	.47±.09	.89±.04	.18±.10	.29±.13	.35±.11	.31±.10	.29±.14	.93±.09	.42±.17	.91±.10	.16±.16	.24±.20	.32±.17	.24±.22
+*MASK*	*.63±.13*	*.64±.13*	*.62±.07*	*.86±.06*	*.83±.08*	*.84±.04*	*.78±.04*	*.56±.07*	*.56±.21*	*.62±.25*	*.53±.16*	*.87±.11*	*.81±.14*	*.83±.07*	*.77±.06*	*.52±.15*
quora-roberta-base	.31±.08	**.99±.02**	.46±.09	.96±.04	.10±.05	.18±.07	.27±.07	.32±.08	.28±.14	.98±.05	.41±.17	.78±.39	.09±.10	.15±.15	.25±.13	.23±.17
+*MASK*	*.63±.12*	*.55±.07*	*.58±.08*	*.83±.04*	*.87±.03*	*.85±.03*	*.78±.03*	*.47±.09*	*.58±.30*	*.47±.18*	*.49±.20*	*.84±.08*	*.88±.08*	*.85±.05*	*.79±.04*	*.41±.16*
quora-roberta-large	.31±.08	.97±.05	.46±.09	.26±.40	.09±.15	.13±.21	.23±.17	.31±.10	.28±.14	.97±.06	.41±.17	.25±.39	.08±.19	.11±.23	.22±.21	.22±.19
+*MASK*	*.40±.20*	*.76±.37*	*.40±.10*	*.22±.34*	*.29±.44*	*.25±.38*	*.30±.26*	*.48±.08*	*.35±.25*	*.73±.41*	*.32±.20*	*.23±.36*	*.29±.44*	*.25±.39*	*.30±.28*	*.42±.14*
ms-marco-MiniLM-L-6-v2	.33±.09	.91±.05	.48±.09	.87±.06	.25±.11	.37±.12	.41±.11	.30±.09	.29±.15	.88±.12	.42±.17	.89±.10	.23±.15	.34±.16	.39±.14	.21±.18
+*MASK*	*.63±.14*	*.62±.13*	*.60±.08*	*.85±.06*	*.84±.07*	*.84±.03*	*.78±.03*	*.55±.07*	*.55±.24*	*.57±.26*	*.49±.20*	*.86±.11*	*.83±.12*	*.83±.05*	*.77±.05*	*.50±.15*
ms-marco-MiniLM-L-12-v2	.33±.09	.79±.07	.46±.09	.81±.07	.35±.09	.49±.09	.49±.07	.24±.09	.30±.17	.78±.17	.41±.18	.82±.16	.34±.15	.47±.16	.48±.12	.20±.16
+*MASK*	*.58±.11*	*.58±.13*	*.56±.05*	*.83±.06*	*.81±.09*	*.82±.05*	*.75±.04*	*.48±.05*	*.47±.19*	*.53±.25*	*.45±.18*	*.83±.11*	*.80±.12*	*.81±.08*	*.74±.06*	*.44±.09*
quora-distilroberta-base	.31±.08	.97±.06	.46±.09	.18±.36	.08±.18	.11±.23	.22±.18	.25±.10	.28±.14	.98±.05	.42±.17	.19±.38	.07±.20	.09±.23	.21±.21	.16±.20
+*MASK*	*.39±.20*	*.84±.31*	*.43±.11*	*.16±.32*	*.20±.39*	*.18±.35*	*.27±.25*	*.34±.10*	*.34±.19*	*.81±.37*	*.36±.21*	*.17±.34*	*.20±.40*	*.18±.36*	*.28±.28*	*.30±.18*
qnli-electra-base	.33±.10	.45±.12	.36±.08	.74±.08	.63±.12	**.67±.08**	.58±.07	.04±.11	.31±.18	.49±.18	.34±.16	.78±.14	.64±.14	**.68±.09**	**.60±.11**	.07±.18
+*MASK*	*.41±.12*	*.36±.12*	*.35±.07*	*.74±.08*	*.77±.14*	*.74±.08*	*.63±.08*	*.07±.08*	*.40±.23*	*.38±.19*	*.32±.11*	*.77±.14*	*.78±.16*	*.75±.10*	*.64±.12*	*.11±.14*
qnli-distilroberta-base	.31±.09	.50±.18	.35±.10	.73±.07	.53±.18	.60±.11	.53±.08	.05±.06	.30±.18	.48±.19	.32±.14	.75±.13	.53±.19	.59±.14	.54±.12	.02±.10
+*MASK*	*.46±.24*	*.31±.15*	*.30±.11*	*.73±.07*	*.77±.16*	*.74±.08*	*.61±.06*	*.13±.09*	*.32±.27*	*.26±.15*	*.24±.13*	*.75±.12*	*.78±.19*	*.74±.11*	*.62±.10*	*.15±.13*

Table E.2: TRiC evaluation using various SBERT models on Subtask 1 and Subtask 2. Results are presented for each model using pre-trained models and the +MASK setting (*italic*). For Subtask 1, precision (PR), recall (RE), and Weighted -F1 scores (F1) are reported for both label 0 (i.e., different topics) and label 1 (i.e., roughly identical topics). For Subtask 2, Spearman correlation (SP) is reported on the overall set of instances. The reported metrics include standard deviations (±) across the 10 Test splits for comparative analysis. The superior performance for each metric between pre-trained models is highlighted in **bold**. Results for both Test and OOV Test sets are provided for completeness.