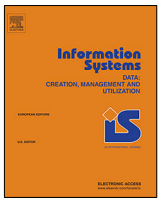
Information Systems 110 (2022) 102107

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is



Transparent assessment of information quality of online reviews using formal argumentation theory

Davide Ceolin [a,*], Giuseppe Primiero [b], Michael Soprano [c], Jan Wielemaker [d]

[a] Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
[b] University of Milan, Milan, Italy
[c] University of Udine, Udine, Italy
[d] SWI-Prolog Solutions b.v., Amsterdam, The Netherlands

ARTICLE INFO

Article history:
Received 29 October 2021
Received in revised form 9 July 2022
Accepted 22 July 2022
Available online 29 July 2022
Recommended by Yannis Manolopoulos

Dataset link: https://github.com/davideceolin/FAReviews

Keywords:
Argumentation reasoning
Information quality
Online reviews

ABSTRACT

Review scores collect users' opinions in a simple and intuitive manner. However, review scores are also easily manipulable, hence they are often accompanied by explanations. A substantial amount of research has been devoted to ascertaining the quality of reviews, to identify the most useful and authentic scores through explanation analysis. In this paper, we advance the state of the art in review quality analysis. We introduce a rating system to identify review arguments and to define an appropriate weighted semantics through formal argumentation theory. We introduce an algorithm to construct a corresponding graph, based on a selection of weighted arguments, their semantic distance, and the supported ratings. We also provide an algorithm to identify the model of such an argumentation graph, maximizing the overall weight of the admitted nodes and edges. We evaluate these contributions on the Amazon review dataset by McAuley et al. (2015), by comparing the results of our argumentation assessment with the upvotes received by the reviews. Also, we deepen the evaluation by crowdsourcing a multidimensional assessment of reviews and comparing it to the argumentation assessment. Lastly, we perform a user study to evaluate the explainability of our method, i.e., to test whether the automated method we use to assess reviews is understandable by humans. Our method achieves two goals: (1) it identifies reviews that are considered useful, comprehensible, and complete by online users, and does so in an unsupervised manner, and (2) it provides an explanation of quality assessments.

1. Introduction

Online reviews can be a valuable source of information, as they allow users to gain from the experience of others who have expressed their opinion about the next product to buy or room to book. Opinions provided by Web users are useful insofar as those of higher quality can be identified, and those that can be characterized as low quality, e.g., for reasons of irrelevance, bias, incompleteness, and so on, can be dismissed. Over the past years, research has characterized reviews' trustworthiness in several ways: user reputation and quality assessment are among them. However, while reviews are about specific products or services, they often express multifaceted views on the target object. To assess the quality and trustworthiness of a review, it is important to understand which arguments it provides, their strengths, and about which aspects of a target product they provide positive or negative evidence. This is an important and novel step in the review analysis process that focuses on the quality of the review itself. This step aims at identifying those reviews whose quality is worth the reader's time. It is, indeed, challenging to check the veracity of reviews, and thus several mechanisms are put in place by reviewing aggregation sites to enhance review truthfulness (e.g., by verifying whether the reviewer purchased the product/service). Even so, a review might be meaningless or useless for a given user, e.g., when it touches minor aspects, or when the opinion it captures is not well-argued. For this reason, we analyze arguments in reviews as a means to assess their quality. This method could be beneficial to several e-commerce activities which intend to identify the highest quality reviews to propose to their users when they scrutinize the next item to buy or service to book. Also, argument-based quality assessment can be useful to improve the methods used to aggregate product ratings, and to improve transparency when these methods are fully automatized e.g., by Artificial Intelligence (AI) and Machine Learning (ML) technologies. The overall rating of a given product might be based on the weighted average of the ratings received,

* Corresponding author.
E-mail address: davide.ceolin@cwi.nl (D. Ceolin).
URL: https://www.cwi.nl/people/davide-ceolin (J. Wielemaker).

https://doi.org/10.1016/j.is.2022.102107

and may take into account in such weight also the argumentative strength and quality of the reviews collected (e.g., possibly giving a lower weight to the low-quality reviews).

In fact, reviews are a means for users to express their opinions on a given product or service. Reviews can be seen in the form of ratings-descriptions pairs. Such form of review, common in many e-commerce sites, indicates a rating (often in a $1-5$ Likert scale) for the quality of a given target product, enriched with textual descriptions motivating the score. We hypothesize that the textual description of a review can provide one or more arguments to support the corresponding Likert scale rating given. Therefore, we use argumentation reasoning to analyze these textual descriptions. Formal Argumentation implements argumentation theory, which is the interdisciplinary study of how conclusions can be reached from premises through logical reasoning: in this formal setting, arguments are the atomic unit of analysis and the scope of the theory is that of analyzing the complex graph resulting from relations between them, considering whether an argument attacks or supports another argument, and to identify which argument survives. Hence, we analyze descriptions within reviews to identify arguments that support the corresponding scores. Arguments are identified through natural language processing of such descriptions and ranked according to their importance using the textRank algorithm [1]. The quality of the descriptions is quantified through a readability measure (e.g., [2]). We formulate, implement, and evaluate a rating system based on formal argumentation theory which collects such sets of pairs when they share a given argument but offer opposing ratings. Once arguments are mined and weighted, we identify a graph of attacks and supports between these arguments referring to the same item. When the numerical ratings of the review they refer to differ, we identify an attack between arguments. We implement the formal rating system developed through a logical reasoner which allows us to identify which arguments (and thus, which reviews) resist the attacks (in the case of disagreeing reviews). We hypothesize that these resulting reviews are those of higher quality. The above analysis addresses the following research questions:

R1: Given a set of reviews about the same product, can argumentation help assess review quality?

R2: Which quality aspects does argumentation emphasize?

R3: Can argumentation be used to explain the review quality assessment process to humans?

In order to implement this argumentation-based rating system, we make use of mature Artificial Intelligence components (e.g., for NLP analysis) as well as ad-hoc developed ones (e.g., the logical reasoner). The novelty of the approach is also in the resulting pipeline. Such a pipeline is created in order to label reviews according to their argument strength. The labeling is then evaluated against the number of upvotes received by reviews (RQ1). Also, we inspect a sample of reviews using a crowdsourcing task meant to obtain a deeper analysis of the quality of such reviews according to different quality aspects (e.g., informativeness) (RQ2). Lastly, since the resulting pipeline is meant to construct an argumentation graph for each product and the review evaluation is based on the reasoning performed on such graph, we test whether the argumentation graph can actually be used to explain the results to a human (RQ3). This article is based on [3] and extends it in three directions: (1) it extensively evaluates the impact of the use of different readability measures on the performance obtained; (2) it extends the crowdsourced dataset by collecting novel assessments so to balance the number of evaluated reviews across the classes determined by the number of stars; and (3) it revises the theoretical foundations of the framework accordingly and extends the related work section.

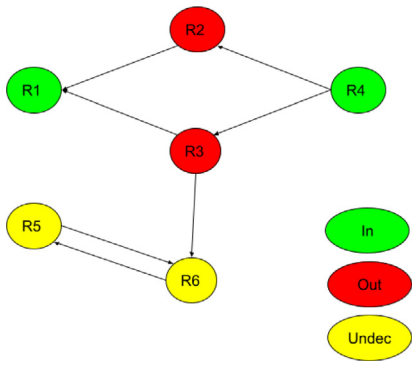


Fig. 1. Example of labeling of reviews following the standard argumentation theory adopted. R1 and R4 are labeled as *in* because either all their attackers are out (R1) or they do not have any attacker (R4); R2 and R3 are labeled as *out* because their attacker is *in*. R5 and R6 have only attackers *undec* or *out*, and are thus labeled as *undec*.

The rest of this paper is structured as follows. In Section 2 we first provide some informal preliminaries, then develop a preferential argumentation framework. In Section 3 we describe the experimental settings we adopt. In Sections Section 4, 5, and 6 we present our approaches to RQ1, 2, and 3, and the related results. We discuss the three evaluations in Section 7. In Section 8 we present related work, and in Section 9 we conclude.

2. Weight-based preferential rating systems

We propose a formal semantics of value-based argumentation that extends the model of Baroni et al. [4] to describe the conflict and support dynamics between tokens within a set of reviews of a given product. In formal argumentation theory, such dynamics is formulated within a graph structure where nodes represent arguments and edges are attacks among them. We interpret nodes of the graph as reviews, requiring that reviews occurring in the same graph refer to at least one common feature of the product under evaluation. Edges of the graph express the attack relation between two reviews assigning different scores to the feature in common. The direction of the attack is given by the relevance of the tokens and the values of the reviews.

The semantics of the graph is defined by a standard formal argumentation theory labeling function on vertices:

1. A review is labeled *in* when all its attackers are *out*;
2. A review is labeled *out* when at least one of its attackers is *in*;
3. A review is labeled *undec* if not all its attackers are *out* and no attacker is *in*.

Fig. 1 illustrates this semantics through an example. This semantics uses a scoring system for tokens within reviews generated from natural language processing which allows translating reviews and their relations in a graph construction algorithm. To this aim, tokens are grouped first using K-means clustering on their distance: two reviews with disagreeing ratings attack each other when they share tokens belonging to the same cluster; among the two, we select only the attack which has a higher value based on the relevance of the token and quality of the review, meaning that only the attack from the more relevant review is considered in the graph construction; the weight of the corresponding edge is based on the semantic distance between tokens. Grouping reviews per token allows us to obtain a coherent set of reviews, identifying positive and negative features of the same aspects of a given product. The clusters of tokens are actually

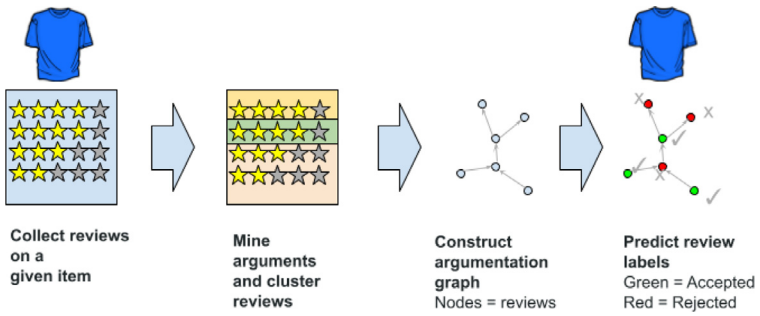


Fig. 2. Review argumentation analysis pipeline. The first step consists of identifying all the reviews related to a given product and of mining their arguments and clustering the reviews according to the semantic distance of the corresponding arguments. The second step consists of constructing the argumentation graph based on the identified arguments and their weights. The third step consists of solving the argumentation graph and labeling the reviews as accepted or rejected.

meant to represent the topics covered by the reviews analyzed. Since reviews represent opinions regarding a given item, we encode attacks between reviews only when their semantic distance is low, i.e., they refer to the same aspects of the item. When their semantic distance is high, we infer that the two reviews refer to different aspects of the item and, thus, the corresponding opinions are treated separately. For example, one opinion might focus on the color of a sweater, the other one on the fabric. If, according to our algorithm, the two tokens are semantically distant, the two reviews are not compared to each other because the corresponding ratings might legitimately disagree. Fig. 2 illustrates the whole pipeline. The first step (argument mining and review clustering) is represented by the first block.

The primary relation between reviews is thus that of attack based on the ordering of the tokens weights: an attacking review must always have a higher score than the attacked one. On the other hand, the relation of support between arguments is represented only indirectly: an argument supports another argument when it attacks at least one of its attackers. Hence, when a valid (*in*) review (i.e. one which has no attackers) attacks another review, it defeats it. This represents the second step of the high-level view of our AI pipeline: arguments from clustered reviews are used to construct the argumentation graph of a given product (see the second block of Fig. 2). Lastly, following [4], we identify which reviews can be accepted based on their arguments surviving the debate, see the last block of Fig. 2.

We now provide formal details of the above informal description, starting with a definition of review. Here we identify all the elements that we need to reason on review arguments. Therefore, our definition of review includes the list of tokens that are mentioned in the review, along with their relevance. Also the review score is represented. Lastly, we make use of a semantic distance measure among tokens and a quality value for the review.

Definition 1 (*Review*)**.** A review $\mathscr{R}_i(t)$ by an agent $i \in \mathscr{A}$ on a target product t is construed as:

1. A list of tokens: $\mathscr{T} = \{\phi_1; ...; \phi_n\}$, expressing the features of relevance of the product t. This list of tokens is meant to characterize which aspects of the product are emphasized and referred to by the review;
2. A relevance value $r(\phi_i) \in [0, 1]$ for each token ϕ_i, expressing the importance of the feature in the overall evaluation of the product t. Since some tokens might be more important than others in determining the review score, we estimate this importance through heuristics and record it through this relevance value;
3. A score $sc(\mathscr{R}_i) = \{1, 2, 3, 4, 5\}$. This score represents the numerical Likert scale score provided by the review;
4. A quality value $v(\mathscr{R}_i) \in [0, 1]$. This value represents the quality value of the review, where 1 indicates the highest possible value.

Provided a set of reviews $\{\mathscr{R}\}$, we collect all those with the same target object t and denote them as $\{\mathscr{R}(t)\}$. The target object is what the review assesses. The list of tokens $\mathscr{T}$ collects the elements characterizing the review content on the target product: for example, on the target "shoes", tokens could be "sole", "upper", but also "comfortable for long walks". The relevance value $r(\phi_i)$ quantifies the importance of token ϕ_i within the review itself. This is a *de facto* value function from tokens to real positive numbers. The score $\mathscr{R}_i$ is the value attributed to the object. We represent the score as an integer from 1 to 5, as is done in many review systems. We currently consider different values as opposing, and do not consider the absolute difference between them (e.g. treating the difference between $| sc(\mathscr{R}_i) - sc(\mathscr{R}_j) |>| sc(\mathscr{R}_i) - sc(\mathscr{R}_k) |$). A quality value is used to weight those reviews which will eventually enter into an attack relation, see below Definition 2, together with the relevance value of all tokens of interest. Quality can be assessed by means of diverse metrics and by looking at diverse aspects (the generic and informal definition of information quality we refer to is "fit for purpose"). In our setting, we consider, in particular, the readability of the review to be an important aspect because:

1. it quantifies how easily a reader might consume it, and
2. it might provide a proxy for the quality of the information it contains.

In our experiments, we use the following readability measures: Flesch–Kincaid Reading Ease, Flesch–Kincaid Grade Level, Automated Readability Index (see [5]), Dale–Chall (see [6]), Simple Measure of Gobbledygook (SMOG, see [7]), Coleman–Liau Index (see [8]), Forcast (see [9]), and we compare their impact on performance.

Tokens within the same cluster are those on which reviews' attacks are defined. The underlying assumption here is that reviews showing semantically distant tokens might be considered incomparable. Hence, we cluster all reviews in $\{\mathscr{R}(t)\}$ sharing the target object t by semantic distance $sem\_dist(\phi_i, \phi_j)$ of their tokens. Using the semantic distance measure, we identify the distances between each pair of tokens ϕ_i, ϕ_j, and we cluster them. We then consider only the attacks that we identify between tokens that belong to the same cluster, which we express as $C(\phi_i) = C(\phi_j)$ where C returns the cluster id for a given token ϕ_i. Different semantic distance measures can be employed at this stage. We refer the reader to the work of Harispe et al. [10] for an exhaustive overview of the field. In our experiments, we use the Word Mover's distance [11] because the tokens we are comparing are often composed of multiple words, and this measure allows us to measure the distance between (short) documents.

We, therefore, define attacks as follows:

Definition 2 (*Attack*)**.** Review $\mathscr{R}_i$ attacks review $\mathscr{R}_j$ with the weight w, denoted as

$\mathscr{R}_i \rightarrow_w \mathscr{R}_j$

if and only if

1. $\mathscr{R}_i(t) = \mathscr{R}_j(t)$;
2. $sc(\mathscr{R}_i) \neq sc(\mathscr{R}_j)$;
3. $\exists \phi_i \in \mathscr{R}_i(t), \phi_j \in \mathscr{R}_j(t)$ such that $C(\phi_i) = C(\phi_j)$;
4. $w = (r(\phi_i) \cdot v(\mathscr{R}_i)) > w' = (r(\phi_j) \cdot v(\mathscr{R}_j))$.

According to the definition above a review attacks another one with the weight w if and only if:

1. they are about the same target object;
2. their score is different (as mentioned above: we make at this point no granular distinction between differences in scores);
3. they have at least one token each occurring in the same cluster, based on their semantic distance;
4. the value w attached to the attack is the highest value obtained by computing for each review the relevance of the token by its quality; this weight w determines the direction of the attack, discarding the attack from the review with lower weight to the review with higher weight; and it also allows to order attacks within a rating system by their strength.

These considerations allow for diverse strategies to be employed in establishing a rating system. A rating system can now be built as a set of reviews and attacks between them, ordered according to a preference relation based on their weights.

Definition 3 (*Rating System*)**.** A rating system is a tuple

$RS := \langle \{\mathscr{R}(t)\}, R^-, \leq \rangle$

where

1. $\{\mathscr{R}(t)\}$ is a list of reviews on target t;
2. $R^- \subseteq \{R(t)\} \times \{R(t)\}$ is a binary relation of attack between reviews, such that $(\mathscr{R}_i, \mathscr{R}_j) \in R^-$ iff $\mathscr{R}_i \rightarrow_w \mathscr{R}_j$;
3. $\leq \subseteq R^- \times R'^-$ is a preference relation such that $R^- \leq R'^-$ if and only if $R^- : \mathscr{R}_i \rightarrow_w \mathscr{R}_j$, $R'^- : \mathscr{R}_k \rightarrow_{w'} \mathscr{R}_l$, $w > w'$ with possibly $j = k$.

According to this Definition, a rating system contains:

1. a set of reviews on the same target,
2. equipped with a set of attack relations,
3. ordered based on their weights.

We now define several strategies to establish the attack relations actually included in any given rating system. These attack strategies are all legitimate strategies we can implement following the elements and strategies described above. In our evaluation, we implement Definition 5, but we provide here a (partial) overview of the possible strategies that it is possible to implement with the elements we identify.

Definition 4 (*Full Attack Strategy*)**.** $\forall R^-, R^- \in RS$.

The Full Attack Strategy includes every well-defined attack relation in the graph, i.e. any review attacks any other review with a different score with which it shares a token within the same semantic distance cluster and which has a lower weight computed as the relevance of the token and quality value of the review.

From this general case, we define a pruning strategy on the number of attack relations, aiming at simplifying the reasoning on the argumentation graph, while removing the less influential attacks:

Definition 5 (*Pruning*)**.** $R^- \in RS$ iff $\exists R'^-.R^- \leq R'^-$ for some $R'^- : \mathscr{R}_i \rightarrow_{w'} \mathscr{R}_j$ and $w' > n$, for some value n.

By Pruning, we remove from the rating system the (set of) attack(s) with a weight lower than a given value n. By the definition of weight, this reflects the intuition that one removes those attacks based on the reviews having a different score on tokens of low relevance, or on semantically distant tokens (i.e. attacks among reviews that express different views on possibly incomparable aspects of the product), removing attacks with a weight under a certain value, e.g. falling within the last percentile.

We now define the labeling of a rating system:

Definition 6 (*Labeling*)**.** Given a rating system *RS*

- $\{\mathscr{S}(t)\} \subseteq \{\mathscr{R}(t)\}$ is conflict-free iff there are no $\mathscr{R}_i, \mathscr{R}_j \in \{\mathscr{S}(t)\}$ such that $(\mathscr{R}_i, \mathscr{R}_j) \in R^-$;
- A review $\mathscr{R}_i \in \{\mathscr{R}(t)\}$ is supported by $\{\mathscr{S}(t)\} \subseteq \{\mathscr{R}(t)\}$ iff for any $\mathscr{R}_j \in \{\mathscr{R}(t)\}$ such that $(\mathscr{R}_j, \mathscr{R}_i) \in R^-$, it exists $\mathscr{R}_k \in \{\mathscr{R}(t)\}$ such that $(\mathscr{R}_k, \mathscr{R}_j) \in R^-$;
- A review $\mathscr{R}_i \in \{\mathscr{R}(t)\}$ is defeated by $\{\mathscr{S}(t)\} \subseteq \{\mathscr{R}(t)\}$ if and only if it $\exists \mathscr{R}_j \in \{\mathscr{S}(t)\}$ such that $(\mathscr{R}_j, \mathscr{R}_i) \in R^-$ and $\mathscr{R}_j$ is supported by $\{\mathscr{S}(t)\}$;
- A review $\mathscr{R}_i \in \{\mathscr{R}(t)\}$ which is neither supported nor defeated is undecided.

A conflict-free *RS* is possible if and only if every review has the same score for every token ϕ_i within a given cluster of semantic distance, i.e., if all the reviews in the same cluster have the same Likert-scale value. The notion of support of a review by a rating system expresses the idea that the score of that review for the given (cluster of) topic(s) is endorsed because any other review attacking it (i.e. with higher weight produced by token relevance and quality) has at least one more review which attacks it. The defeat of a review by a rating system expresses the dual idea that the score of that review for the given (cluster of) topic(s) is rejected: i.e. the review presenting that score has an attacker within the rating system, i.e. another review with a higher weight attacking it. Finally, an undecided review is one that presents a high expected variance on its usefulness in establishing the score of the product.

Standard semantics types from Formal Argumentation Theory are here adapted for our Rating System:

Definition 7 (*Semantics*)**.** Given a rating system *RS*

- A conflict free set $\{\mathscr{S}(t)\} \subseteq \{\mathscr{R}(t)\}$ is admissible iff each $\mathscr{R}_i \in \{\mathscr{S}(t)\}$ is supported by $\{\mathscr{S}(t)\}$;
- A preferred extension is an admissible subset of $\{\mathscr{R}(t)\}$ maximal w.r.t. set-inclusion and weight ordering;
- An admissible $\{\mathscr{S}(t)\} \subseteq \{\mathscr{R}(t)\}$ is a complete extension iff each review supported by $\{\mathscr{R}(t)\}$ is in $\{\mathscr{S}(t)\}$;
- The least (with respect to set inclusion) complete extension is the grounded extension.
- The most (with respect to the weight of supported reviews) complete extension is the weighted complete extension.

In particular, in the following, we look for the model which maximizes the number of *in*-nodes with higher weight, i.e. we concentrate on the preferred extension of an admissible set of reviews for a given rating system. The rationale behind this choice is to extract from a given rating system the "ideal review", or the one composed by the evaluation of the largest number of tokens that are rated highest within the system. This allows for preserving completeness of analysis of the product and quality of the

reviews. The analysis of other semantics, and their justifications, is left for further work.

Example. We consider here a simple example of a rating system based on a partial formulation of the graph represented in Fig. 1. Consider the following:

- $\mathscr{R}(t) = \{R_1, R_2, R_3, R_4\}$;
- $\mathscr{T} = \{\phi_1, \phi_2, \phi_3, \phi_4\}$;
- $sc(R_1) = sc(R_4) = 1$, $sc(R_2) = sc(R_3) = 4$: the four reviews are pairwise agreeing with each other;
- $\phi_i \in R_i, \forall i$
- $r(\phi_1) = 0.2$, $r(\phi_2) = 0.1$, $r(\phi_3) = 0.5$, $r(\phi_4) = 0.6$;
- $v(R_1) = 0.3$, $v(R_2) = 0.4$, $v(R_3) = 0.6$, $v(R_4) = 0.7$;
- $C(\phi_i) = C(\phi_j), \forall i, j$;
- $w_1 = 0.2 \cdot 0.3 = 0.06$, $w_2 = 0.1 \cdot 0.4 = 0.04$, $w_3 = 0.5 \cdot 0.6 = 0.3$, $w_4 = 0.6 \cdot 0.7 = 0.42$;
- $\leq = \{w_2 < w_1 < w_3 < w_4\}$: R_4 is the review with the highest weight (w_4), thus it attacks all disagreeing reviews, R_1 and R_3. Vice-versa, R_2 is the review having the lowest weight, and it is thus attacked by R_1 and R_3;
- $R^- = \{R_1 \rightarrow_{0.06} R_2, R_3 \rightarrow_{0.3} R_2, R_4 \rightarrow_{0.42} R_3, R_4 \rightarrow_{0.42} R_1\}$.

In this rating system:

- with the Full Attack strategy we preserve the full set R^-; R_1, R_4 are accepted because these are either not attacked (R_4), or their attackers are defeated (R_1). R_2, R_3 are defeated by R_4. $\mathscr{S}(t) = \{R_1, R_4\}$ is a complete extension of $\mathscr{R}(t)$;
- with the Pruning strategy we might want to remove the attack with the lowest weight, namely $R_1 \rightarrow_{0.06} R_2$; note that in this case, the pruning would not have any effect, while it would in absence of $R_4 \rightarrow_{0.42} R_3$, because then R_1 would survive and R_2 would not.

3. Experimental setting

We describe the implementation of the above framework and the dataset adopted.

3.1. Implementation

As mentioned above, Fig. 2 provides an overview of our pipeline.[1] We describe the pipeline implementation as follows.

Feature extraction. Given a set of reviews for product target t, we extract:

1. The set of textual tokens in such reviews to use as the set of tokens $\mathscr{T}$ and their importance in the text $r(\phi_i)$ for each token $\phi_i \in \mathscr{T}$. Textual tokens are estimated using the Spacy library, their importance is estimated through the pyTextRank library implementing the TextRank algorithm, i.e., computing the PageRank of the tokens in the review based on their textual dependency. The only cleaning step we do apply here is stop word removal, using the nltk package. We will consider refining the cleaning and tokenization process as a future extension.
2. The readability scores of the review to use as a proxy for $sc(\mathscr{R}_i)$; again we use the Spacy library and, in particular, the Spacy-readability extension.

Argumentation graph building. We proceed as follows:

1. Build the semantic distance matrix of all the tokens in the reviews of that product from each $sem\_dist(\phi_i, \phi_j)$. Since we are comparing tokens that are composed of multiple words, we use here a document distance measure. We use the Word Mover's distance [11] implemented in Gensim [12] to this aim. This measure derives the distance between two given documents (in this case, two tokens) based on the semantic distance between the pairs of words that compose them. Word semantic distance is computed by means of embedding-based distance;
2. cluster tokens according to their semantic distance. We use K-means and we identify the optimal number of clusters using the silhouette method. We proceed as follows:
   (a) We run the method systematically for each product.
   (b) We identify the reviews about the given product and extract the corresponding tokens.
   (c) The text distance between tokens is measured using the Word Mover's distance [11]. This provides us with a distance matrix between the tokens (M).
   (d) We iterate k in $(2, min(10, |M|))$, so that the highest value chosen for k is the minimum between 10 and the cardinality of M.
   (e) We run K-Means using the above k values, and compute the silhouette score every time, using the klearn Python package.
   (f) We select the value of k that maximizes the silhouette score.

   Thus, the value of k varies per product.
3. represent an argumentation graph as a NetworkX Directed graph where: (1) nodes represent reviews; and (2) links represent attacks. Reviews attack all other reviews with a lighter score that share the same token and disagree on the rating.

Graph solution. In order to identify the models of the graph, we implement a SWISH Prolog-based solver also available as a standalone service accessed via a customized extension of the Python Prolog Pengines library.[2]

3.2. Dataset

We evaluate the above model on the Amazon Review Dataset [13], in particular on the Amazon Fashion 5-core dataset, which consists of 3,176 unique reviews provided by 406 users about 31 products. We use this dataset for the evaluation of RQ1. For the evaluation of RQ2, our goal is to analyze via crowdsourcing a stratified sample with the same number of reviews per number of stars assigned. Thus, we need to extend the sample of reviews on which to apply crowdsourcing. However, the 5-core dataset does not contain a sufficient number of reviews to this aim.

In more detail, the 5-core dataset is a subset of the complete Amazon Review Dataset [13] where each of the users and items has 5 reviews each. However, after further inspection, we noticed that the dataset contains just a fraction of the subset of 5-cores user/item pairs. To evaluate RQ2 we firstly rebuilt the whole 5-core dataset. The Amazon Review Dataset consists of 883,636 reviews (871,502 after duplicate removal) provided by 746,352 users about 185,241 products. To rebuild the whole 5-core dataset, we sample 5 reviews for each product ASIN code provided by 5 different authors. Each user/product pair must be unique across the whole dataset. In this way, there are 5 different authors of reviews for every product and 5 reviews provided by the same author for 5 different products, thus complying with the 5-core assumption. Hence, the final 5-core dataset is made of 148,588 reviews about 29,958 products, which corresponds to

[1] Source code available at: https://github.com/davideceolin/FAReviews.

[2] https://swish.swi-prolog.org/p/argue.swinb

16.81% of the original Amazon Review Dataset. We also extract the metadata for each product. Thus, to evaluate RQ2 we augment the sample used in the first version of this work [3] by sampling reviews from the whole 5-core dataset to obtain the same number of reviews per each number of stars/upvotes. The final augmented sample balanced along the number of stars/upvotes is composed of 670 reviews which corresponds to 0.45% of the original Amazon Review Dataset.

For each review, the dataset reports:

- the id of the review author;
- the timestamp and the text of the review;
- the id of the product reviewed;
- the rating given to the product (on a 1-5 Likert scale);
- the number of upvotes a given review received. Note that users can only indicate whether they found a given review useful, not the opposite.

For each product, the dataset reports:

- the name of the product;
- the description of the product;
- the subcategory of the product;
- various attributes about its size, weight, dimensions, etc.

3.3. Argumentation graph building example

Reviews and their argumentation graph are compared for explainability. Details of the graph construction process are given below in Section 4. Here we provide an example:

> Review 1: 'We have used these inserts for years. They provide great support.' (5 stars)
>
> Review 2: 'This is my 6th pair and they are the best thing ever for my plantar fasciitis and resultant neuromas. Unfortunately, the ones I ordered from SmartDestination must be seconds as they kill my feet. The hard plastic insert rubs on the outside edges of my feet. I am unable to exchange them as I waited one day too late to use them in my walking shoes.' (2 stars).

The two reviews have no textual token in common, however, some of their tokens are semantically related. For example, 'these inserts' (Review 1) and 'hard plastic insert' are semantically close enough to belong to the same cluster. This means that we capture an attack between the two, from Review 1 (readability 102.5 using the Flesch–Kincaid Reading Ease measure) to Review 2 (readability 73.44 using the same Flesch–Kincaid Reading Ease measure). The weight of the attack is given by the weight of the attacking node, i.e., by the weight of the token with higher weight. The document distance is computed after stop words removal (the above tokens are 'hard plastic insert' and 'inserts'). This process is repeated with all the tokens shared between two reviews and with all the review pair combinations for a given product. Fig. 3 illustrates this example. We can see that Review 1 (R1) is depicted with a larger circle so as to signify its higher readability score. Also, the token identified shows a higher importance in the review than the token on Review 2. Thus, Review 1 attacks (and defeats, in this small example) Review 2.

4. RQ1 - Review quality assessment evaluation

We consider here the ability of our system to discriminate reviews' quality.

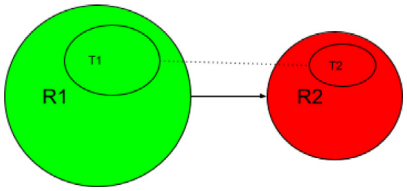


Fig. 3. Review 1 is depicted with a larger circle as to signify its larger weight. Similarly, token T1 has a larger importance in R1 than T2 has in R2. Consequently, after having identified a semantic link between the two tokens, we identify an attack from R1 to R2. R2 is defeated, following the semantics of the argumentation framework described in Section 2.

4.1. Baselines and evaluation settings

We created two baselines:

Unsupervised (K-Means). We extract a set of basic textual features from the reviews (text length; number of words; number of tokens, computed using the Spacy Python library; textual readability, computed using the Flesch–Kincaid Reading Ease measure) and we cluster them using the K-Means algorithm with $K = 2$. Note that here K-means is used to label reviews as 'in' or 'out', i.e., as a method to accept and reject reviews alternative to argumentation reasoning. This use of K-means is thus totally different from the use we make in Section 3.1. There, we use it to cluster an arbitrary number of tokens according to their semantics. In that case, we needed to identify the ideal k parameter given the tokens observed, and we used the silhouette method to identify it. Here, instead, we have a fixed number of target labels ('in', 'out') and thus we use a fixed value for k.

Supervised (SVC). Using the same features as above, we split the dataset and use the first 30% of reviews to train a Support Vector Classifier to classify the remaining 70%. To allow a fair comparison between the three methods, we convert the number of upvotes into two buckets, to mimic the classification obtained with our method. We provide three variations on this, with thresholds at 1, 5, and 10 upvotes: therefore, we consider as accepted (*in*) those reviews that received at least 1, 5, or 10 upvotes respectively.

We evaluate our framework under three different settings:

Argumentation Framework We adopt the dataset described in Section 3.1.

Argumentation Framework Weighted We adopt the dataset described in Section 3.1, but we apply a decaying function to the number of upvotes based on their age. The decaying function we use is

$$w(x) = \frac{t_{max} - t_x}{t_{max} - t_{min}}$$

where t_{max} and t_{min} are the highest and lowest timestamps in the dataset; t_x is the timestamp of review x. Since the argumentation framework result is compared with a snapshot of the upvotes collected at a given time, this decaying function compensates for the fact that the older reviews had a higher chance to get upvotes than the younger ones.

Argumentation Framework Weighted (Upvotes>0) Since votes can only be up and not down, we cannot tell whether zero-votes reviews deserve zero votes or negative votes. For this reason, we also checked the algorithm's performance when considering reviews that received at least one upvote.

Table 1
Average number of upvotes received by the reviews in each class. The average of upvotes in the class should be maximized, the average in the *out* class minimized. In Arg. framework weighted, a temporal decaying function (see Section 4.1) is used. In Arg. framework weighted (> 0 upvotes), we consider reviews with at least 1 upvote (see Table 6). We report the results obtained with all readability measures considered.

Method	Out	In
Arg. framework	2.3	0.5
Arg. framework weighted	**0.0**	0.4
Arg. Fram. Weigh. (>0 upvotes)	**0.0**	4.2
Arg. framework (SMOG)	0	5.9
Arg. framework weighted (SMOG)	0	5.7
Arg. framework weighted (>0 upvotes, SMOG)	0	5.7
Arg. framework (Dale–Chall)	19	5.7
Arg. framework weighted (Dale–Chall)	5.7	5.7
Arg. framework weighted (>0 upvotes, Dale–Chall)	5.7	5.7
Arg. framework (Forcast)	0	5.8
Arg. framework weighted (Forcast)	0	5.7
Arg. framework weighted (>0 upvotes, Forcast)	0	5.7
Arg. framework (ARI)	19	5.7
Arg. framework weighted (ARI)	0	5.7
Arg. framework weighted (>0 upvotes, ARI)	0	5.7
Arg. framework (Flesch–Kincaid Reading Ease)	19	5.7
Arg. framework weighted (Flesch–Kincaid Reading Ease)	0	5.7
Arg. framework weighted (>0 upvotes, Flesch–Kincaid Reading Ease)	0	5.7
Arg. framework (Coleman–Liau Index)	35	5.7
Arg. framework weighted (Coleman–Liau Index)	0	5.7
Arg. framework weighted (>0 upvotes, Coleman–Liau Index)	0	5.7
Unsupervised (K-Means)	2.5	0.3
Supervised (SVC) @1	**0.0**	5.7
Supervised (SVC) @5	0.2	10.1
Supervised (SVC) @10	0.3	**17.7**

4.2. Results

We run the above algorithm and we obtain a classification of product reviews as *in* or *out*. No review is classified as *undecided*. Table 1 shows the average number and sum of upvotes that the review in a given class received. For example, the reviews that are labeled as *out* (i.e., rejected) by the weighted version of our framework got, on average, 5.7 upvotes, and reviews classified as *out* got on average 0.0 when using the SMOG readability measure. For each readability measure, we apply the argumentation framework, and we compare its performance with the number of votes received by the reviews in each class (*in* and out reviews). We also apply a temporal-based weighting factor to the votes, and we also check the performance of the algorithm on the reviews that received at least one vote. We apply these strategies to all results obtained with the diverse readability measures and we discuss these strategies further in Section 7. Besides the choice of the specific readability measure, also the choice of the k value of the K-means algorithm might affect the performance of our framework. We explain in Section 3.1 that we use the silhouette method to identify the best value for k for each product (and, thus, for each group of reviews). In order to analyze the sensitivity of this parameter, we test the algorithm performance with three more values for k, two static, and one dynamic. As for the static values of k, we select 1 and 2, because the number of tokens per group of reviews is highly variable, so we choose values that aim at being applicable to all groups of reviews we analyze. Actually, the use of k = 1 allows us to evaluate the use of clustering at all since with k = 1 we actually do not cluster reviews and thus record attacks between all tokens (see Section 2). The additional dynamic strategy sets k equal to half of the cardinality of the set of identified tokens. In our framework, once we identify a group of reviews per product, we cluster the corresponding tokens based on their distance (using K-means) and then weight the tokens using a combination of their textRank centrality and the readability of the review they are extracted from. Based on this, we identify attacks between reviews and, lastly, label reviews as accepted or rejected (*in* or *out*) using argumentation theory. We saw above that the choice of the readability measure has a small effect on the overall performance, but still, some readability measures yield better results than others. Thus, we pick one of the best-performing readability measures (Automated Readability Index) as well as one of the worst-performing ones (Flesch–Kincaid Reading Ease) and we check how the performance varies with different values for k. Table 3 reports the corresponding results, which are discussed in Section 7. We already anticipate that the impact of the choice of the value for k is significant and that our strategy based on the silhouette method is the best performing one. Also, the resulting labeling identifies groups of reviews that have a significantly different scores across each quality dimension. However, a such significant difference is only relative: this means that when k is set to 1, 2, or equal to half of the cardinality of the set of tokens, for any dimension, the score for reviews *in* is significantly higher then the score of reviews *out* resulting *in*. However, the scores of the first set are not necessarily higher than zero, and the scores of the second set are not necessarily lower than zero. These conclusions have been derived by applying a Mann–Whitney Test at 95% confidence level.

We considered the possibility of computing the precision and recall of our method. However, precision and recall imply the existence of negative samples, while upvotes are only positive values. Artificially introducing a threshold to split reviews into positive and negative items would be possibly misleading. A "one-size-fits-all" would hardly work in this case: such a threshold could have to vary per product or product type and could have to take into account also temporal aspects. For instance, less popular products could receive fewer reviews and have a smaller chance to get upvotes. Thus, their threshold should be lower than that of popular products. At the same time, the rareness of reviews alone cannot be considered a sufficient reason to set the bar low: those few reviews could get few upvotes because of their poor quality. Although our method outputs discrete labelings that could be matched with positive and negative labeling of reviews, a corresponding counterpart based on the number of upvotes is not available (as it is unclear how many upvotes a review

Table 2
Sum of upvotes received by the reviews in each class. The average of upvotes in the class should be maximized, the average in the *out* class minimized. In Arg. framework weighted, a temporal decaying function (see Section 4.1) is used. In Arg. framework weighted (>0 upvotes), we consider reviews with at least 1 upvote (see Table 6). We report the results obtained with all the readability measures considered.

Method	Out	In
Arg. Fram. (SMOG)	**0**	**1740**
Arg. framework weighted (SMOG)	0	1690
Arg. framework weighted (>0 upvotes, SMOG)	0	1690
Arg. framework (Dale–Chall)	76	1664
Arg. framework weighted (Dale–Chall)	23	1667
Arg. framework weighted (>0 upvotes, Dale–Chall)	23	1667
Arg. framework (Forcast)	0	1740
Arg. framework weighted (Forcast)	0	1690
Arg. framework weighted (>0 upvotes, Forcast)	0	1690
Arg. framework (ARI)	0	1740
Arg. framework weighted (ARI)	0	1690
Arg. framework weighted (>0 upvotes, ARI)	0	1690
Arg. framework (Flesch–Kincaid Reading Ease)	76	1664
Arg. framework weighted (Flesch–Kincaid Reading Ease)	23	1667
Arg. framework weighted (>0 upvotes, Flesch–Kincaid Reading Ease)	23	1667
Arg. framework (Flesch–Kincaid Grade Level)	76	1664
Arg. framework weighted (Flesch–Kincaid Grade Level)	23	1667
Arg. framework weighted (>0 upvotes, Flesch–Kincaid Grade Level)	23	1667
Arg. framework (Coleman–Liau Index)	70	1670
Arg. framework weighted (Coleman–Liau Index)	11	1678
Arg. framework weighted (>0 upvotes, Coleman–Liau Index)	11	1678
Unsupervised (K-Means)	662	962
Supervised (SVC) @1	62	1109
Supervised (SVC) @5	344	827
Supervised (SVC) @10	674	497

Table 3
Count of the upvotes received by the reviews classified as 'in' and 'out' by our algorithm with diverse values for the k parameter of the K-means algorithm. We can see that the best performing combinations are those based on our method, that uses the silhouette method. The performance of the other methods are identical. The actual labeling resulting from applying the different values for k is not always the same, but since these differences affect reviews with 0 votes, the resulting counts are identical.

Readability Measure	k = 1 (in,out)	k = 2 (in,out)	k = $\|tokens\|/2$ (in,out)	k = silhouette(tokens) (in,out)
Automated Readability Index	(1635,105)	(1635,105)	(1635,105)	(1740,0)
Flesch–Kincaid Reading Ease	(1635,105)	(1635,105)	(1635,105)	(1664,76)

needs to be considered of high quality, and if it is possible to set such a number at all). Therefore, we limit the comparison with the baseline approaches to Table 1. With these considerations in mind, to allow a comparison between our method and SVC, we still introduce the use of thresholds to convert the multivalued classification of SVC into binary values but we make use of different thresholds exactly because the ideal value for this is not given. Thus, we use three thresholds, 1, 5, and 10 (the mean number of upvotes received by a review in the ground truth is 0.55, median 0). We deepen these considerations in Section 7 (see Table 2).

5. RQ2 - Multidimensional review quality assessment

The evaluation of the argumentation theory-based review assessment by correlation with upvotes uses the latter as the only ground truth provided in the dataset at our disposal, but they also show important limitations. First, upvotes collect only positive votes: if a review did not get a high number of upvotes, it could be either of low or average quality. Second, the semantics of upvotes is rather vague and broad: since they are the only means for readers to express their endorsement, they can capture appreciation in a too broad sense. Third, upvotes might depend on the order in which reviews are exposed to users and their ages. We extend our analysis of the quality of reviews to obtain a more thorough and detailed gold standard. We crowdsource answers to questions regarding quality aspects of a significant number of reviews, as detailed below.

5.1. Crowdsourcing setting

We collect 670 reviews by randomly selecting one of the products reviewed at a time and then drawing one of its reviews until we balance the number of reviews collected per review score value, i.e., the number of stars assigned. This leads to an amount of 134 reviews for each score. We ask each worker to evaluate the quality of 10 reviews and each review is evaluated by 5 workers. Workers are located in the US, and the tasks (which are rewarded 0.9$) are performed through Amazon Mechanical Turk[3], using the Crowd_Frame platform [14].

Task description. We present the worker with a product description as provided in the Amazon dataset. Then, we present the review, and we ask the worker to assess the review on a 5-level Likert scale (from −2, completely disagree, to +2, completely agree), across the following quality dimensions:

Truthfulness: measures the overall truthfulness and trustworthiness of the review.

Reliability: the review is considered reliable, as opposed to reporting unreliable information. *Example (label: +2 Completely agree): "They fit great, look great, are quite comfortable and are just what I was looking for!".*

[3] http://mturk.com

Neutrality: the review is expressed in objective terms, as opposed to resulting subjective or biased. *Example (label: −2 Completely disagree): "Love them!!"*

Comprehensibility: the review is comprehensible/understandable/readable as opposed to difficult to understand. *Example (label: +2 Completely agree): "They run big. Order a full size smaller".*

Precision: the review is precise/specific, as opposed to vague. *Example (label: +2 Completely agree): They run big. Order a full size smaller.*

Completeness: the review is complete as opposed to partial. *Example (label: +2 Completely agree): "I actually have 3 pairs of these trainers. They are very comfortable, there is a neoprene sleeve that goes around your ankle that makes them the most comfortable for me compared to normal athletic shoes. They run a little narrow — for me this is perfect, but you may want to round up on the size or try on in the store first if your feet are on the wider side".*

Informativeness: The review allows deriving useful information as opposed to well-known facts and/or tautologies. *Example (label: +1 Agree): "Love these shoes! Needed new running shoes and these are perfect. Light weight and fit great!"*

The above dimensions are based on previous work on multidimensional quality assessment [15]. However, with reviews, it is very hard for the workers to determine the truthfulness of information because they need to assess the authenticity of the review itself, which is often subjective. So, we adapt the quality dimensions from the literature to represent more subjective aspects like reliability.

5.2. Results

Assessments were collected and we checked whether the scores in any of the evaluated dimensions showed a correlation with the *in-out* evaluation of the reviews by our algorithm. Since our classification consists of two labels only, while the crowdsourced data are multidimensional and finer-grained, we performed a set of analyses at diverse levels of aggregation, starting from splitting the reviews into *in* and *out*, obtaining the results summarized below. We also test the existence of a correlation between our labels and the crowdsourced assessments for each combination of labels obtained with the seven readability measures mentioned above:

- a χ^2 on the two sets of review scores: no significant difference is identified. We analyzed the assessments obtained with all seven readability measures, and no assessment showed a significant difference;
- a Mann–Whitney test on the average score per dimension: no significant difference between the two sets of reviews is identified. Again, this holds for all readability measures;
- t-test and Mann–Whitney test when comparing the raw scores on each dimension show a significant difference between the distribution of the information quality scores and that of the review assessments, for all readability measures.

Then, we aggregate the scores in two ([−2,0],[1,2]) and three ([-2,-1], [0],[1,2]):

- a χ^2 test on the two sets of reviews identifies a significant difference between the information quality scores and the argumentation-based assessment when a two-scores bucket is used. This holds for all readability measures, except for SMOG;
- a Mann–Whitney test on the average score per dimension identifies a significant difference in the distribution of scores, independent of which readability measure is employed;
- at 90% confidence, a significant difference is identified in the distribution of some of the information quality dimensions when using the three-valued aggregations. Table 4 shows which readability measures are able to support the identification of argumentation-based assessments that allow dividing reviews into two different sets that are significantly different according to a specific quality dimension. The table shows the presence of a significant difference when this is identified either via a t-test or a Mann–Whitney test. We can observe that ARI, followed by Forcast, are the readability measures performing better in this sense.

In other words, when the crowdsourced scores are expressed on a coarse scale (three-valued, in particular), when using specific readability measures (ARI and Forcast, in particular) our classification identifies two sets of reviews, where those labeled as *in* have higher chance to be of a different quality than those labeled as *out*. When using the ARI readability measure, this holds for seven quality dimensions. These kinds of quality assessments are thus more refined than the count of upvotes, and here the choice of the readability measure does actually affect the results. We will investigate in the future whether the use of crowd worker profile information could help in weighing the readability scores. In fact, some readability measures (e.g., Flesch Kincaid Grade Level) are meant to classify the text difficulty according to the expected level of complexity that people with different levels of education can understand. By considering background information, we could thus filter or weigh the workers' assessments so to better align them with the argumentation-based assessments, depending on the readability measure used. Since the readability score plays a role in the argumentation framework, those results might just be linked to the use of those scores. However, all readability scores show a weak correlation with crowdsourced comprehensibility. Thus, the identification of the reviews with higher quality (e.g., comprehensibility, completeness) can be attributed to the whole framework.

6. RQ3 - Explainability evaluation

We run an explorative questionnaire[4] to evaluate whether our approach provides informative explanations on the decision taken about the reviews (*in*/*out* outcome). We select two reviews about the same product, one accepted, and one rejected by our system. We show the argumentation graph on which the judgment is based and we ask the respondent whether the graph helps in understanding the underlying reasoning using a 1-5 Likert scale. Users can provide additional feedback. Table 5 shows the distribution of the 31 anonymous responses received, while Fig. 4 shows an example graph proposed to the participants, together with the corresponding description.

According to these results, the argumentation graph does indeed help in explaining the outcome. Since the outcomes vary from 'poorly informative' (1) 'to very informative' (5), the results are explanatory on both reviews (although for review 2 the signal is stronger). An important aspect of consideration as a possible limitation is that in argumentation-based reasoning arguments are valid until attacked and this translates into reviews accepted when they are not attacked.

[4] The questionnaire is available at https://forms.gle/srGJpGyYBzWd9RTaA.

Table 4
Comparison of the ability of the different readability measures to support the identification of argumentation reasoning-based assessments that imply two significantly different groups of reviews, according to the information quality dimensions considered. The table reports an 'x' when either a Mann–Whitney or a t-test lead to H_0 acceptance at 90% confidence level.

	Dale-Chall	Coleman Liau Index	F.-K. Grade Level	F.-K. Reading Ease	SMOG	Forcast	ARI
Informativeness		x				x	x
Reliability		x				x	x
Overall Truthfulness							x
Completeness	x	x	x			x	x
Comprehensibility	x	x	x			x	x
Neutrality	x		x			x	x
Precision	x		x			x	x

Table 5
Distribution of the answers regarding the helpfulness.

Informativeness	1 (Poorly Informative)	2	3	4	5 (Very Informative)
Review 1 (accepted)	0	6	11	12	2
Review 2 (rejected)	0	1	7	15	8

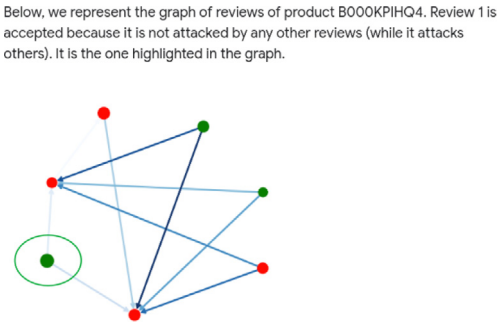


Fig. 4. Example graph and the corresponding description. Nodes represent reviews, (green *in*, red *out*) for the argument-based review classification, arrows represent attacks, their shade expresses semantic distance (darker shade indicates lower distance, lighter shade indicates higher distance). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

7. Discussion

We now discuss the results related to each research question.

7.1. RQ1 - Given a set of reviews about the same product, can argumentation reasoning help assess review quality?

Our method (especially in the improved version that applies a temporal-based weighting on the number of votes obtained) identifies two clusters of reviews where those labeled as in have a higher chance of having more upvotes than those out. Also, the method identifies the majority of the reviews that received upvotes. The choice of the readability measure employed in our framework has a little impact on the performance.

The first difference between the unsupervised approach and the proposed argumentation framework concerns labeling. The results reported in Table 1 assume arbitrarily that one of the two classes predicted by the K-means method equals the *out* class, the other the *in* class. However, we do not have any means to label the classes in this respect. So, while the performance of the two methods looks similar when considering the averages in Table 1, this may not be the case. For most of the remaining performance reported in Table 1, our method outperforms K-means. Also, as we can see from Table 1, the choice of the readability measure slightly impacts the performance, but the highest improvement comes from the use of a temporal-based weighting factor. The best-performing versions of the argumentation theory-based classifications are those using the SMOG, Forcast, Automated Readability Index (ARI). Also, regarding the choice of the value of k for the K-means algorithm, our strategy that employs the silhouette method yields the best performance. This is due to the fact that this strategy dynamically adapts to the number and distribution of tokens extracted per review. While this strategy is computationally more expensive than setting a fixed threshold, it allows to better label as accepted those reviews that actually received upvotes. The supervised approaches are those showing the best performance in terms of the distribution of the average number of upvotes. Supervised approaches focus on identifying the peculiarities of reviews that hint at their upvotes. They do so at the dataset level, they make use of labeled data (number of upvotes per review) and can identify those reviews that meet these criteria. These methods achieve high accuracy of the number of upvotes estimated for a given review. However, they do so at the expense of a significant amount of upvotes missed, as the right table of Table 1 shows. Measuring performance as precision and recall would have meant comparing our method on the mere ability to identify reviews having at least n upvotes for an arbitrary threshold n (this step is necessary to transform the number of upvotes in the ground truth into binary values comparable with our classification). This introduces the problem of deciding if and how to determine whether a review can be considered as of high or low quality based on the number of upvotes received or of additional parameters. This goes beyond our goals. The correct threshold should depend on the number of reviews received by a given product, etc. We use thresholds to transform SVC in binary outcomes, though, because of the quantitative nature of SVC. SVC predicts the number of upvotes received by a review. Setting a threshold introduces the mentioned limitations but, in this case, performance would have been measured in terms of error of the number of upvotes predicted. Thus, thresholds mainly reduce the granularity of such metrics but necessarily introduce some error: reviews that got n upvotes for $0 < n <$ threshold are labeled as out, thus affecting the performance reported in the right table of Table 1. Also, when the performance of the supervised methods is good, it comes with limitations:

Need for training data. Being supervised, SVC craves for labeled data; in production, the system might be affected by the cold start problem. In general, this approach will require part of the data to be evaluated and to be used for training purposes. Also, while we used an arbitrary 30%–70% ratio between training and

test sets, the use of this approach might require the additional step of identifying the best performing ratio. This additional step is not required by our unsupervised approach.

Arbitrary parameters. When comparing the two methods, we had to convert the estimated number of upvotes into two classes. This is arbitrary because it corresponds to answering a question like "how many upvotes does a review need to receive to be accepted?". This has led to testing the three different parameters.

Lack of explanations. The method is meant to estimate the number of upvotes received by each review. However, when deciding whether to consider a given review or not based on such estimates, it is important to understand how such reasoning was performed. Inspection on the importance would require additional efforts.

These limitations are not shown by our method, which is unsupervised and explainable. Also, from the diverse evaluation settings, we learned the following lessons.

Lesson learned 1: Time matters. When inspecting the reviews in the *out* class, the high average is due to just one review labeled as out, despite having received 35 upvotes. This is the oldest review of that product; 6 more reviews, received about 6 years later, had 0 upvotes. Given that these newer reviews got a lower chance to get an upvote because they are more recent, we discounted the number of upvotes based on the age of the review. This improves the system performance (see Table 1).

Lesson learned 2: Non-attacked reviews should not necessarily be accepted. In formal argumentation theory, arguments are accepted until they are defeated. However, not yet attacked reviews could get zero upvotes for a variety of reasons (e.g., they are off-topic). On a long-tail distributed dataset, this affects the results obtained. This is the reason why the reviews classified *in* have a low average number of upvotes. As shown in the third row of Table 1, the performance on the reviews with at least one upvote is higher. Table 6 provides an overview of the number of reviews per class.

Lesson learned 3: The choice of the specific readability measure has a minor impact on performance. It is true that the use of some readability measures leads to a low performance, like in the case of the Coleman–Liau and of the Flesch–Kincaid Reading Ease, that lead to a set of reviews labeled as 'out' having an average higher number of votes than those 'in'. However, this is mitigated in all the cases by the use of temporal weights on the number of votes. This effect of the choice of the readability measure becomes secondary as it can be mitigated. The extensive analyses performed on the use of diverse readability measures allows for this conclusion to be made. This insight provides an additional step towards the 'actionability' of the pipeline transparency: if the user intends to tweak the pipeline, they now can understand what are the implications of the choice of different readability measures.

Lesson learned 4: Dynamically setting the value k of the K-means algorithm using the silhouette method yields the best results. We tested the performance of our method with different values of k, both setting it statically to 1 and 2 and dynamically to half of the cardinality of the set of the tokens identified. The performance of our framework with these values for k is significantly lower than the performance of the same framework when k is selected dynamically using the silhouette method. When looking at the impact of k on the quality assessments per dimension, the value of k does not introduce major differences. Also when k is set statically, we can discriminate reviews across different quality dimensions only in relative terms. This means that the resulting labeling differentiates groups of reviews that have relatively different scores, however, the groups of reviews identified by such labeling does not clearly identify one group of "good" (scores > 0 for a given dimension) and one of "bad" (scores < 0 for the same dimension) reviews. When k is chosen using the silhouette method, we obtain this distinction for some quality dimensions.

Table 6
Number of reviews classified as in and out, split on the number of upvotes.

Class	in	out
Reviews with 0 upvotes	2,706	14
Reviews with at least 1 upvote	288	1

7.2. RQ2 - Which quality aspects does argumentation reasoning emphasize?

The classification performed by our argumentation framework is correlated with the quality of the reviews, mostly with their comprehensibility and completeness. ARI is the readability measure that determines argumentation-based assessments with the highest correlation to quality dimensions. As already pointed out in Section 5, the readability scores alone would not be able to point out the reviews having higher overall truthfulness, as all readability scores show a very weak correlation with comprehensibility assessments. This result has a twofold consequence. First, it supports the argumentation-based approach and the need for logical reasoning to be performed on top of the ranked arguments to obtain labeling that correlates with overall truthfulness and comprehensibility. Second, it points out other quality aspects that we might consider in future extensions of our framework. E.g., completeness might be correlated to the number of *in* arguments in a review.

Also here we learned an important lesson.

Lesson learned 5: Granularity and semantics matter. While quality is subjective and contextual, it is also possible to define which aspects of quality we are interested in. This is important to allow a more precise understanding of the argumentation outcome. Also, the current implementation of the framework provides a three-valued assessment and, as expected, correlation with crowdsourced ratings emerges only when these are aggregated in buckets. Future extensions of the framework might consider a fine-grained representation of acceptance/rejection of arguments.

7.3. RQ3 - Can argumentation reasoning be used to explain the review quality assessment process to humans?

According to the exploratory study described in Section 6, argumentation graphs are useful to explain review assessment. The study was meant to provide a first indication about the hypothesis that argumentation graphs are useful to explain review assessment. The respondents agreed with this concept: 45,2% of them rated informativeness at level 4 or 5 (very informative) for the first question, 73,6% for the second. This will be further explored in the future. "How to better represent attack weights?" and "which level of complexity users can handle?" are examples of questions we will tackle.

8. Related work

This work falls within the growing family of weighted argumentation frameworks extending standard Dung's setting, including Preferential Argumentation Frameworks [16–18] and Value-based Argumentation Frameworks [19,20]. A specific approach is represented by systems defining preferences based on weighted attacks, see [21], establishing that some inconsistencies are tolerated in the set of arguments, provided that the sum of the weights

of attacks does not exceed a given value. Weights can be used to provide a total order of attacks, see [22]. This approach can be generalized in several ways: in [23] a different way of relaxing the admissibility condition and strengthening the notion of defense is presented; in [24] different selections on extensions based on the order of weights are proposed. Our work also relies on an ordering on weighted attacks, essential differences being that:

1. the definition of weights is given by the semantic distance between tokens;
2. the clustering of attacks is based on weights;
3. the pruning of the graph is based on the order, as distinct from the selection of the model based on the maximization of the weight of accepted arguments.

Research on the assessment of the quality and credibility of product reviews has focused mostly on linguistic aspects, e.g., based on readability and linguistic errors [25–28]. While such approaches can be a source of inspiration for future extensions, the main difference with our approach is the combination of such linguistic aspects with argumentation reasoning. A similar extension can be obtained by looking into credibility factors, as in [29]. Lastly, [30] looks for a junction between natural language processing and argumentation reasoning. While it classifies more thoroughly the diverse tokens as different kinds of arguments, it does so semi-automatically, while we take an automatic unsupervised approach. We intend, however, to improve our argument mining step. Currently, we naïvely treat all the tokens in a review as potential indicators of arguments, and we weigh their importance. However, we will better refine argument identification in the future. To this aim, we will refer to the large body of literature related to argument mining, which is summarized, for instance, in the review of Lawrence and Reed [31], the one of Lippi and Torroni [32], and the one of Moens [33]. While preserving our focus on mining arguments from product reviews, we will better clarify the identification of arguments. Refining argument characterization is another aspect we intend to improve in the future. In particular, the work of Hinton and Wagemans [34] provides an interesting classification of argument types called "periodic table of arguments" which we will consider operationalizing in the future. Identifying and reasoning on argument types is important to understand the strength and weight of arguments in debates. The work of Čyras et al. [35] surveys methods to determine the strength of argumentation-based methods, focusing on the use of argumentation for explainability. When improving the identification and classification of arguments, however, we will have to consider that the type of data we focus on is semi-structured. Reviews are composed of a numerical rating and a textual explanation, which makes them quite specific compared to the textual data analyzed in the literature about argumentation. The work of Stab et al. [36] on mining arguments on heterogeneous sources, and the work of Michel et al. [37], who employ a combination of natural language processing and of knowledge graphs to reason on arguments, both provide interesting methods that we will consider adapting in the aforementioned extensions.

The work of Cocarascu et al. [38] is also relevant to us. They make use of argumentation reasoning to aggregate and explain movie reviews. Applying their approach to product reviews seems to be an interesting future work direction. Similarly, Briguez et al. [39] apply argumentation reasoning on movies, in particular in the context of supporting recommendations. While they do not explicitly focus on reviews, our work could be used to support review recommenders, and thus their approach provides valuable insights in light of possible future extensions.

Regarding the crowdsourced assessment of online information, we refer the reader to the work of Roitero et al. [40], although their focus is on political statements, and their assessment is mono-dimensional. They further extend their approach by collecting multidimensional assessments in [41]. A similar multidimensional approach on the assessment of Web documents involving experts is also adopted in [15].

Lastly, concerning the visual representation of argumentation graphs for explanation and exploration purposes, relevant work is the one of Moser and Mercer [42], who develop a graphical model for biomedical purposes. While their use case implies a much more complex argumentation graph, we intend to explore this technique in the future.

9. Conclusion and future work

This paper presents a framework for classifying reviews' quality based on a combination of NLP and argumentation reasoning. We evaluate the framework on a real-world dataset showing that this approach partly outperforms baseline unsupervised and supervised approaches, while also providing explainable results. We also investigated the impact of the choice of readability measures in the performance evaluation, and we observed that these have little impact on the ability of the framework to identify useful reviews.

A deeper analysis of the quality of the reviews based on crowdsourcing highlights that the argumentation framework is actually capable of identifying those reviews that the users perceive as more comprehensible and complete. Also, three-level scoring of reviews across multiple quality dimensions is revealed to be the ideal level of granularity. The readability measures that show a higher ability to discriminate between high- and low-quality reviews are ARI and Forcast. While comprehensibility is correlated with the scores obtained with argumentation reasoning, it is also important to note that no readability score shows a correlation with the comprehensibility scores. Thus, the readability measures alone are not sufficient to explain the performance obtained by argumentation reasoning. Also, a deeper investigation on the impact of the value of the k parameter in the K-means algorithm we employ shows that setting k using the silhouette method consistently provides positive results.

We also run a user study that confirms the ability of argumentation graphs of providing useful explanations. This argumentation-based framework represents a first step towards a reliable and transparent assessment of the quality of online opinions.

We foresee several future developments for this work. Firstly, the framework should be extended by discounting the weight of the review and its attacks considering the temporal aspect (e.g., using weight $w(x)$ of Section 7). Secondly, the model could account for the different semantics of nodes *in* and *out* to prevent novel reviews to be automatically *in*. Thirdly, we will improve the identification of the arguments among the review tokens. For example, we intend to explore the use of knowledge graphs to enrich product reviews and better model their arguments. This requires an additional step of entity linking, to link identified entities in reviews to items in well-known knowledge bases (e.g., Wikidata). However, once such a link will be established, it will be possible to leverage explicit semantic relations between items in the knowledge bases to identify similar tokens in reviews. Fourthly, other variations of the definition of attack might be investigated, including for example a novel computation of the weight which might consider the difference between weights of

the two reviews involved, rather than simply accounting for the weight of the stronger review. This would have the immediate effect of discerning more sensibly between different attacks with the same attacker. Lastly, we intend to investigate in-depth the existence of correlations between workers' profiles in the crowdsourced setting and the performance of the different readability measures.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The code we use is published on GitHub, openly available at https://github.com/davideceolin/FAReviews. We analyze data previously published by McAuley et al. (https://jmcauley.ucsd.edu/data/amazon/).

Acknowledgments

This work is partially supported by The Credibility Coalition, partially by the Project "Departments of Excellence 2018-2022" awarded to the Department of Philosophy "Piero Martinetti" of the University of Milan, partially by the PRIN2020 Grant no. 2020SSKZ7R of the Italian Ministry of University and Research (MUR), and partially by the "Eye of the Beholder" Project (project number 027.020.G15) of the Netherlands eScience Center.

References

[1] R. Mihalcea, P. Tarau, TextRank: Bringing order into text, in: Proceedings of EMNLP, ACL, 2004, pp. 404–411.
[2] J. Kincaid, R. Fishburne, R. Rogers, B. Chissom, Derivation of New Readability Formulas for Navy Enlisted Personnel. Research Branch Report 8–75, Tech. rep., Chief of Naval Technical Training: Naval Air Station Memphis, 1975.
[3] D. Ceolin, G. Primiero, J. Wielemaker, M. Soprano, Assessing the quality of online reviews using formal argumentation theory, in: M. Brambilla, R. Chbeir, F. Frasincar, I. Manolescu (Eds.), Web Engineering, Springer International Publishing, Cham, 2021, pp. 71–87.
[4] P. Baroni, M. Caminada, M. Giacomin, Abstract argumentation frameworks and their semantics, in: P. Baroni, D. Gabbay, M. Giacomin (Eds.), Handbook of Formal Argumentation, College Publications, 2018, pp. 159–236.
[5] J.P. Kincaid, R.P. Fishburne Jr., R.L. Rogers, B.S. Chissom, Derivation Of New Readability Formulas (Automated Readability Index, Fog Count And Flesch Reading Ease Formula) For Navy Enlisted Personnel, Tech. rep., Research Branch Report 8–75. Chief of Naval Technical Training: Naval Air Station Memphis, 1975.
[6] E. Dale, J.S. Chall, A formula for predicting readability, Educ. Res. Bull. 27 (1) (1948) 11–28.
[7] G.H. McLaughlin, SMOG grading – a new readability formula, J. Reading 8 (1969) 639–646.
[8] M. Coleman, T.L. Liau, A computer readability formula designed for machine scoring, J. Appl. Psychol. 60 (1975) 283–284.
[9] J. Caylor, Methodologies for Determining Reading Requirements of Military Occupational Specialists, Technical report, Human Resources Research Organization, 1973.
[10] S. Harispe, S. Janaqi, J. Montmain, Semantic Similarity from Natural Language and Ontology Analysis, Morgan Claypool Publishers, Williston, VT, USA, 2015, http://dx.doi.org/10.2200/S00639ED1V01Y201504HLT027.
[11] M.J. Kusner, Y. Sun, N.I. Kolkin, K.Q. Weinberger, From word embeddings to document distances, in: Proceedings of ICML, JMLR.org, 2015, pp. 957–966.
[12] R. Řehůřek, P. Sojka, Software framework for topic modelling with large corpora, in: Proceedings of NLPFrameworks Workshop, ELRA, 2010, pp. 45–50.
[13] J.J. McAuley, C. Targett, Q. Shi, A. van den Hengel, Image-based recommendations on styles and substitutes, in: Proceedings of SIGIR, ACM, 2015, pp. 43–52.
[14] M. Soprano, K. Roitero, F. Bombassei De Bona, S. Mizzaro, Crowd_Frame: a simple and complete framework to deploy complex crowdsourcing tasks off-the-shelf, in: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, WSDM '22, Association for Computing Machinery, 2022, pp. 1605–1608, http://dx.doi.org/10.1145/3488560.3502182.
[15] D. Ceolin, J. Noordegraaf, L. Aroyo, Capturing the ineffable: Collecting, analysing, and automating web document quality assessments, in: Proceedings of EKAW, Springer, 2016, pp. 83–97.
[16] L. Amgoud, C. Cayrol, A reasoning model based on the production of acceptable arguments, Ann. Math. Artif. Intell. 34 (2002) 197–215.
[17] S. Modgil, Reasoning about preferences in argumentation frameworks, Artificial Intelligence 173 (9) (2009) 901–934.
[18] L. Amgoud, S. Vesic, Two roles of preferences in argumentation frameworks, in: Proceedings of ECSQARU, Springer, 2011, pp. 86–97.
[19] T.J.M. Bench-Capon, Value-based argumentation frameworks, in: Proceedings of NMR Workshop, 2002, pp. 443–454.
[20] T.J.M. Bench-Capon, Persuasion in practical argument using value-based argumentation frameworks, J. Logic Comput. 13 (3) (2003) 429–448.
[21] P.E. Dunne, A. Hunter, P. McBurney, S. Parsons, M. Wooldridge, Weighted argument systems: Basic definitions, algorithms, and complexity results, Artificial Intelligence 175 (2) (2011) 457–486.
[22] D.C. Martínez, A.J. García, G.R. Simari, An abstract argumentation framework with varied-strength attacks, in: Proceedings of KR, AAAI Press, 2008, pp. 135–144.
[23] S. Coste-Marquis, S. Konieczny, P. Marquis, M.A. Ouali, Weighted attacks in argumentation frameworks, in: Proceedings of KR, AAAI Press, 2012, pp. 593–597.
[24] S. Coste-Marquis, S. Konieczny, P. Marquis, M.A. Ouali, Selecting extensions in weighted argumentation frameworks, in: Proceedings of COMMA, IOS Press, 2012.
[25] A. Ghose, P.G. Ipeirotis, Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics, IEEE Trans. Knowl. Data Eng. 23 (10) (2011) 1498–1512.
[26] P. Wu, H. Van Der Heijden, N. Korfiatis, The influences of negativity and review quality on the helpfulness of online reviews, in: Proceedings of ICIS, 2011, pp. 3710–3719.
[27] N. Korfiatis, E. García-Bariocanal, S. Sánchez-Alonso, Evaluating content quality and helpfulness of online product reviews: The interplay of review helpfulness vs. review content, Electron. Commer. Res. Appl. 11 (3) (2012) 205–217.
[28] G. Ocampo Diaz, V. Ng, Modeling and prediction of online product review helpfulness: A survey, in: Proceedings of ACL, Vol. 1, ACL, 2018, pp. 698–708.
[29] C.N. Wathen, J. Burkell, Believe it or not: Factors influencing credibility on the web, J. Am. Soc. Inf. Sci. Technol. 53 (2) (2002) 134–144.
[30] A. Wyner, J. Schneider, K. Atkinson, T. Bench-Capon, Semi-automated argumentative analysis of online product reviews, in: Proceedings of COMMA, IOS Press, 2012, pp. 43–50.
[31] J. Lawrence, C. Reed, Argument mining: A survey, Comput. Linguist. 45 (4) (2019) 765–818, http://dx.doi.org/10.1162/coli_a_00364, URL https://aclanthology.org/J19-4006.
[32] M. Lippi, P. Torroni, Argumentation mining: State of the art and emerging trends, ACM Trans. Internet Technol. 16 (2) (2016) http://dx.doi.org/10.1145/2850417.
[33] M.-F. Moens, Argumentation mining: How can a machine acquire common sense and world knowledge? Arg. Comput. 1 (2018) 1–14.
[34] M. Hinton, J. Wagemans, Evaluating reasoning in natural arguments: A procedural approach, Argumentation 36 (2022) 61–84, http://dx.doi.org/10.1007/s10503-021-09555-1.
[35] K. Čyras, A. Rago, E. Albini, P. Baroni, F. Toni, Argumentative XAI: A survey, in: Z.-H. Zhou (Ed.), Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, International Joint Conferences on Artificial Intelligence Organization, 2021, pp. 4392–4399, http://dx.doi.org/10.24963/ijcai.2021/600, Survey Track.
[36] C. Stab, T. Miller, B. Schiller, P. Rai, I. Gurevych, Cross-topic argument mining from heterogeneous sources, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 3664–3674, http://dx.doi.org/10.18653/v1/D18-1402, URL https://aclanthology.org/D18-1402.
[37] F. Michel, F. Gandon, V. Ah-Kane, A. Bobasheva, E. Cabrio, O. Corby, R. Gazzotti, A. Giboin, S. Marro, T. Mayer, M. Simon, S. Villata, M. Winckler, Covid-on-the-web: Knowledge graph and services to advance COVID-19 research, in: ISWC 2020 - 19th International Semantic Web Conference, Athens / Virtual, Greece, 2020, http://dx.doi.org/10.1007/978-3-030-62466-8_19, URL https://hal.archives-ouvertes.fr/hal-02939363.
[38] O. Cocarascu, A. Rago, F. Toni, Extracting dialogical explanations for review aggregations with argumentative dialogical agents, in: AAMAS, 2019, pp. 1261–1269, URL http://dl.acm.org/citation.cfm?id=3331830.

[39] C.E. Briguez, M.C. Budán, C.A. Deagustini, A.G. Maguitman, M. Capobianco, G.R. Simari, Argument-based mixed recommenders and their application to movie suggestion, Expert Syst. Appl. 41 (14) (2014) 6467–6482, http://dx.doi.org/10.1016/j.eswa.2014.03.046, URL https://www.sciencedirect.com/science/article/pii/S0957417414001845.

[40] K. Roitero, M. Soprano, S. Fan, D. Spina, S. Mizzaro, G. Demartini, Can the crowd identify misinformation objectively? The effects of judgment scale and assessor's background, in: Proceedings of SIGIR, ACM, 2020, pp. 439–448.

[41] M. Soprano, K. Roitero, D. La Barbera, D. Ceolin, D. Spina, S. Mizzaro, G. Demartini, The many dimensions of truthfulness: Crowdsourcing misinformation assessments on a multidimensional scale, Inf. Process. Manage. 58 (6) (2021) 102710.

[42] E. Moser, R.E. Mercer, Use of claim graphing and argumentation schemes in biomedical literature: A manual approach to analysis, in: ARGMINING, 2020.