UNIVERSITÀ DEGLI STUDI DI MILANO

- Dipartimento di Informatica e Comunicazione -

SCUOLA DI DOTTORATO IN INFORMATICA
XXIV CICLO

Semantic Data Clouding over the Webs

INF/01 INFORMATICA

Gaia Varese

Supervisor: Prof. Silvana Castano
Assistant supervisor: Dr. Alfio Ferrara

Headmaster of the Ph.D. School: Prof. Ernesto Damiani

Academic Year 2011

to my family

Acknowledgements

The first acknowledgment is dedicated to my supervisor Prof. Silvana Castano, my assistant supervisor Dr. Alfio Ferrara, and my colleague Dr. Stefano Montanelli. Working with them has represented a great opportunity for my professional and personal growth. I would also like to acknowledge the referees, Prof. Isabel F. Cruz, Prof. Carlos A. Heuser, and Prof. Riccardo Torlone, for their comments and attention.

Milano, January 16th, 2012

Abstract

Very often, for business or personal needs, users require to retrieve, in a very fast way, all the available relevant information about a focused *target entity*, in order to take decisions, organize business work, plan future actions. To answer this kind of "entity"-driven user needs, a huge multiplicity of web resources is actually available, coming from the Social Web and related user-centered services (e.g., news publishing, social networks, microblogging systems), from the Semantic Web and related ontologies and knowledge repositories, and from the conventional Web of Documents. The Ph.D. thesis is devoted to define the notion of *in*-cloud and a *semantic clouding approach* for the construction of *in*-clouds that works over the Social Web, the Semantic Web, and the Web of Documents. *in*-clouds are built for a target entity of interest to organize all relevant web resources, modeled as *web data items*, into a graph, on the basis of their level of *prominence* and reciprocal *closeness*. Prominence captures the importance of a web resource within the *in*-cloud, by distinguishing, also in a visual way "a la tag-cloud", how much relevant web resources are with respect to the target entity. The level of closeness between web resources is evaluated using matching and clustering techniques, with the goal of determining how similar web resources are to each other and with respect to the target entity.

Contents

List of Figures

Chapter 1

Introduction

The expectations of users on the quality of the results obtained from searching the web are becoming higher and higher. Very often, for business or personal needs, users require to retrieve, in a very fast way, all the available relevant information about a focused *target entity*, in order to take decisions, organize business work, plan future actions. A target entity is a keyword-based representation of a topic of interest, namely a real-world object/person, an event, a situation, a movie, or any similar subject that can be of interest for the user. To answer this kind of "entity"-driven user needs, a huge multiplicity of web resources is actually available, coming from the Social Web and related user-centered services (e.g., news publishing, social networks, microblogging systems), from the Semantic Web and related ontologies and knowledge repositories, and from the conventional Web of Documents. In order to refer to all such kinds of web, we use the concept of "Webs", to describe the idea that a plurality of different kinds of web is currently available. In particular, the *Web of Documents* can be defined as the set of traditional, typically static, web pages, together with the set of hypertext links connecting them through the World Wide Web [Berners-Lee et al., 1994]; the *Social Web* can be defined as the set of websites that are designed and developed in order to support and foster social interaction, together with the set of social relations that link people through the World Wide Web [Porter, 2008]; the *Semantic Web* can be defined as the set of semantically structured web pages, together with the semantic relations that link them through the World Wide Web [Berners-Lee et al., 2001]. Each kind of web resource is differently structured according to a variety of formats, ranging from short, unstructured, and ready-to-consume news/posts, to well-structured, formal

ontology, and each one can provide unique information for a given target entity. For example, only web resources coming from the Social Web are able to provide subjective information reflecting users opinions or preferences about the target entity, which complement in a useful way the more objective information provided by web resources coming from the other Webs [Easley and Kleinberg, 2010]. To satisfy user expectations, a new generation of web information search techniques has to cope with different requirements: i) the capability to span across multiple Webs, to properly consider the wide variety of available web resources and pieces of knowledge by properly assessing their information contribution nature; ii) the capability to anticipate the user needs by providing a focused but comprehensive set of web resources relevant for the target entity; iii) the capability to semantically organize all the retrieved web resources into an intuitive and coherent structure for the given target entity [Koutrika et al., 2009b].

1.1 Thesis contribution and outline

With respect to this scenario, the Ph.D. thesis is devoted to define the notion of *in*-cloud and a *semantic clouding approach* for the construction of *in*-clouds that work over the Social Web, the Semantic Web, and the Web of Documents [Castano et al., 2010b; Varese, 2011]. *in*-clouds are built for a target entity of interest to organize all the relevant web resources, modeled as *web data items*, into a graph, on the basis of their level of *prominence* and reciprocal *closeness*. Prominence captures the importance of a web resource within the *in*-cloud, by distinguishing, also in a visual way "a la tag-cloud", how much relevant web resources are with respect to the target entity. The level of closeness between web resources is evaluated using matching and clustering techniques, with the goal of determining how similar web resources are to each other and with respect to the target entity.

The thesis is organized as follows. In Chapter 2, we present the state of the art of semantic data clouding, by distinguishing the contributions provided in the field of Social and Semantic Web. In Chapter 3, we introduce our semantic clouding approach and we describe the unified model that is used to represent the different kinds of web resources. Chapter 4 is devoted to present a literature survey of instance matching and record linkage techniques, and to describe all the matching techniques that we have developed to compare heterogeneous web resources. In Chapter 5, we formally define

the concept of *in*-cloud and we describe the techniques that are used for classifying and clouding web resources in order to build *in*-clouds. Experimental results on the proposed semantic clouding approach are discussed in Chapter 6. Finally, in Chapter 7, we give our concluding remarks and we discuss the most interesting directions for future work. Appendix A is devoted to describe the benchmark that we have created in order to evaluate the presented matching techniques.

The Ph.D. activity has been partially related to the EU FP6 BOEMIE (Bootstrapping Ontology Evolution with Multimedia Information Extraction) project, where the described matching techniques have been adopted to compare annotated web pages [Castano et al., 2009a, 2011].

Chapter 2

State of the art

Relevant research work with respect to the Ph.D. thesis regards representation and aggregation techniques proposed in the field of Social and Semantic Web. Thus, in this chapter, we describe the state of the art solutions proposed respectively to work with web resources extracted from the Social Web and from the Semantic Web.

Related work regarding matching techniques are presented separately in Chapter 4, where we also present the matching techniques that we have developed to compare heterogeneous web resources. Such choice is motivated to the fact that matching techniques are involved only in a specific task of our approach, and thus they are treated independently. On the other hand, in this Chapter, we present approaches and techniques that are similar to our semantic data clouding approach as a whole.

2.1 Social Web

The increasing popularity of Social Web and related user-centered services like news publishing, social networks, tagging and microblogging systems, have led to the availability of a huge bulk of messy data, that are mostly characterized by short textual descriptions with poor metadata and a basic structure [Castano et al., 2010c,d]. However, they become an essential source of information, sometimes unique, to answer users' queries about specific events/topics of interest with the goal of providing subjective information reflecting users' opinions/preferences. In this direction, information search and retrieval is moving from traditional information lookup to exploratory

search, defined as the activity of finding and understanding knowledge about a topic of interest by exploiting aggregation and learning of information in a social context [Marchionini, 2006]. However, the research efforts towards the development of solutions for organizing/aggregating Social Web resources according to clouding techniques or similar approaches are still at an initial stage [Kuo et al., 2007; Hernández et al., 2008; Koutrika et al., 2009a]. Existing works are mainly focused on defining techniques and applications for news aggregation and for semantic organization of tags into taxonomies/ontologies. Such approaches are similar to the one proposed in the thesis, because they have a similar objective. However, the existing tools generally work only on a specific category of web resources (e.g., news, web pages). Instead, our approach is able to work with multiple web resources at the same time.

2.1.1 News aggregation

Some interesting work has been done in the field of news aggregation with the aim of providing techniques for their semantic organization and classification. Examples of proposed systems are Relevant News [Bergamaschi et al., 2007], RSS Clusgator System [Li et al., 2007], NewsInEssence [Radev et al., 2005], Velthune [Gullí, 2005], NewsJunkie [Gabrilovich et al., 2004], and QCS [Dunlavy et al., 2003].

Relevant News [Bergamaschi et al., 2007] is a news feed aggregator which automatically groups news related to the same topic published in different newspapers in different days, on the basis of the similarity of their titles.

The RSS Clusgator System [Li et al., 2007] applies a hierarchical clustering algorithm over the retrieved news, in order to better serve the reader in finding the news of interest.

NewsInEssence [Radev et al., 2005] is a system for finding and summarizing clusters of related news articles from multiple sources on the web. The system aims to automatically generate summaries of news events by using a centroid-based summarization technique. It considers the salient terms appearing in each cluster of related documents, and uses these terms to construct the clusters summary. Then, it exploits search-engines available online in order to retrieve news of interest for a user, on the basis of a sample article or of a set of keywords.

The Velthune search-engine [Gullí, 2005] is able to retrieve, index, classify, and

cluster news published using both RSS and Atom formats.

NewsJunkie [Gabrilovich et al., 2004] is a system that personalizes news articles for users by identifying the novelty of stories in the context of stories users have already reviewed.

QCS [Dunlavy et al., 2003] is a software tool and development framework for streamlined IR. The system matches a query to relevant documents of news articles, clusters the resulting subset of documents by topic, and produces a summary for each topic.

2.1.2 Semantic organization of tags into taxonomies/ontologies

In recent years, tagging systems have acquired a great popularity in the field of Social Web. Tagging systems allow users to annotate web resources (e.g., text documents, images, videos, web pages) by associating them a set of tags. Tags are terms arbitrarily chosen by users for their capability to describe the content of web resources. The resulting set of tags and web resources within a tagging system is called *folksonomy*. The popularity of tagging systems is mainly due to their ease of use. In fact, users can easily classify web resources without having any technical knowledge and without being constrained by specific conventions. However, the complete freedom of choosing any term for the annotation of web resources inevitably leads to the generation of messy sets of tags. A lot of research is currently focused on trying to organize folksonomies and associate with them a certain degree of semantics [Begelman et al., 2006; Cattuto et al., 2007; Echarte et al., 2007; Mika, 2007; Specia and Motta, 2007; Tummarello and Morbidoni, 2007], in order to enable semantic resource search.

In the following, we discuss the state of the art in the field of the semantic organization of tags extracted from tagging systems into toaxonomies/ontologies by distinguishing two main categories of approaches dealing with social tagging system management.

Tag classification approaches. These approaches aim at extracting taxonomies or ontologies from folksonomies using some kind of tag classification technique. For example, the approaches presented in [Laniado et al., 2007] and in [Lin et al., 2009] rely on the use of the WordNet lexical dictionary [Miller, 1995] to detect correct relations

between tags.

In [Specia and Motta, 2007], the authors propose a methodology to build an ontology starting from a set of tags by exploiting information harvested from WordNet, Google, Wikipedia and other similar knowledge repositories available in the web. This way, it is possible to automatically detect terminological relations between tags like synonymy or hyponymy to be used for tag classification.

Schmitz [Schmitz, 2006] proposes a probability model to build an ontology from tags extracted from Flickr. Subsumption relations between tags are mined on the basis of the conditional probability between pairs of tags, by considering the number of web resources containing each tag and the number of users who used each tag.

An alternative approach for building a taxonomy starting from a set of tag assignments is presented in [Barla and Bieliková, 2009]. In this work, a parent-child or a sibling relation between each tag and its most frequently co-occurring tag is established. The choice about the kind of relation to consider is taken with the help of WordNet.

Mika [Mika, 2007] provides a model of semantic-social networks for extracting lightweight ontologies from del.icio.us, which exploits co-occurrence information for clustering tags over relevant concepts.

Heymann and Garcia-Molina [Heymann and Garcia-Molina, 2006] propose a method for building a hierarchical taxonomy according to a defined measure of tag centrality in the folksonomy graph.

A similar approach is presented in [Eda et al., 2009], where authors distinguish between subjective tags, which reflect user's ideas about web resources (e.g., "cool", "funny"), and objective tags, which are related to the web resources themselves (e.g., "tutorial", "webtechnology"). Thus, the tag taxonomy is created by taking into account only the objective tags.

A different approach to deal with folksonomy mapping into ontologies is presented in [Echarte et al., 2007]. In this work, authors propose to build an RDF description of a generic folksonomy, where the ontology concepts represent the elements of the folksonomy itself, rather than general concepts.

Similarity-based search approaches. Several contributions deal with the issue of defining similarity-based techniques for social annotations with the goal of improving web resource search and retrieval. A survey of similarity measures for collaborative

tagging systems is provided in [Markines et al., 2009].

Cattuto et al. [Cattuto et al., 2008] propose a method for creating networks of similar web resources. In particular, similarity between web resources is determined by analyzing the tags used for their annotation, their respective TF-IDF value, and their intersection. The TF-IDF value (Term Frequency - Inverse Document Frequency) [Salton and Buckley, 1997] is a measure which is used in information retrieval to evaluate the importance of a term for a specific document in a collection of documents. The importance increases proportionally to the number of times a term appears in the document but is offset by the frequency of the term in the whole collection. Applied to social tagging, the TF-IDF value can be used to evaluate the importance of a tag for a specific web resource, by counting the number of times the tag has been used to annotate such web resource and the number of times the tag has been globally used to annotate other web resources.

In [Tummarello and Morbidoni, 2007], authors propose an application, called DBin, where networks of similar users are created in order to collaboratively build RDFS ontologies over domains of interest starting from the del.icio.us tags.

Similarity techniques exploiting the co-occurrence between tags are described in [Begelman et al., 2006] for tag clustering and in [Sigurbjörnsson and Van Zwol, 2008] to provide meaningful suggestions during the tagging phase of photos in Flickr.

A formal model to enhance the information retrieval functionalities of folksonomies is provided in [Hotho et al., 2006a,b]. In particular, Hotho et al. [Hotho et al., 2006b] propose a method for converting a folksonomy into an undirected weighted graph, used for computing a modified PageRank algorithm, called FolkRank, for ranking query results. In [Hotho et al., 2006a], authors propose to use FolkRank in order to identify the relevance of each web resource, user, and tag, with respect to a specific target web resource, user, and tag.

In [Heymann et al., 2008], authors study the impact that social tagging can have in the traditional web search, analyzing tags in del.icio.us, with respect to the web pages they are associated with.

2.2 Semantic Web

Differently from the Social Web, the Semantic Web is not suitable for users without any technical knowledge, as it requires a specific technical background for both creat-

ing and exploiting the Semantic Web resources, which are characterized by a complex structure and a set of semantic relations with the other Semantic Web resources. In fact, the aim of the Semantic Web is to structurally organize the web resources, giving them a semantics, which consists in the identification of semantic relations between object descriptions, as well as in the recognition of the descriptions which are referred to the same real-world object. The most important attempt in creating such kind of web is the Linked Data project [Berners-Lee et al., 2008; Bizer et al., 2009]. In particular, Linked Data can be defined as a method for exposing, sharing, and connecting pieces of data, information, and knowledge, using URIs and RDF. It is mainly focused on the idea of improving interoperability and aggregation among large data collections already available on the web by linking together descriptions of the same real-world object which are stored in different RDF repositories, such as for example DBLP[1], DBpedia[2], CiteSeer[3], IMDB[4], and Freebase[5]. Moreover, Linked Data is a step beyond the simple availability of data and syntactic compatibility, in that it promotes some important principles in making web resources available and sharable to the Semantic Web community. Such principles are the following: i) all the web resources have to be referenced by a URI; ii) URIs have to be resolvable on the web to RDF descriptions; iii) RDF triples have to be consumed by a new generation of Semantic Web browsers and crawlers.

In this field, works most strictly related to the topic of the thesis are the ones aiming at presenting to the users the Linked Data and, more in general, the Semantic Web, in a more intuitive way. Such approaches can be mainly classified into two different categories, according to the way RDF datasets are visualized and navigated [Deligiannidis et al., 2007]. The first kind of visualization consists in browsing a labeled oriented graph, while the second one consists in displaying RDF properties as browsable facets of a node. However, all such approaches are related to only a limited portion of the huge amount of web resources actually available on the web, that is, the Semantic Web resources, and they do not take into account the web resources originated from user-generated contents like comments, posts, and personal feeds. Instead, our semantic

[1] http://www.informatik.uni-trier.de/~ley/db

[2] http://dbpedia.org

[3] http://citeseerx.ist.psu.edu

[4] http://www.imdb.com

[5] http://www.freebase.com

clouding approach is conceived to consider both Social and Semantic Web resources.

Graph-oriented visualization. Graph-oriented visualization approaches exploit the concept relations in the RDF graph and provide some kind of entity aggregation. For example, in [Hirsch et al., 2009], authors propose a tool for a visual navigation and exploration of Freebase. Given a topic of interest, the tool produces a graph-based representation of it, where nodes are associated with an icon representing the type of the node, as defined in Freebase, and edges are labeled with the name of the corresponding relationships. Moreover, related topics of the same type are combined in aggregated nodes, and a textual description of each selected node is provided.

In [Mirizzi et al., 2010], a tool that helps users in exploring DBpedia is presented, not only via directed links in the RDF dataset, but also via newly discovered knowledge associations and visual navigation. Moreover, it exploits aggregation techniques in order to combine related topics in unified nodes, providing also a textual description of each node.

In [Mutton and Golbeck, 2003], the authors focus on discovering the disconnections in an ontology graph, in order to provide the visualization of smaller graphs, which can be navigated exploiting the topology of the original graph. However, such approach is suited to work only with relatively small ontologies, and it does not scale well for huge dataset as DBpedia.

The approach proposed in [Stuckenschmidt et al., 2004] supports the exploration of large online document repositories, by providing a concept-based visualization (based on clustering techniques) of query results.

LESS [Auer et al., 2010][6] is an approach providing a set of web-based templates to define visual representations of Linked Data. LESS templates may take as input one or more data sources via SPARQL queries. The resulting visual output can be embedded as formatted HTML into web pages or can be produced in form of RSS.

The main drawback of graph-oriented visualization approaches is that they do not scale to large datasets [Frasincar et al., 2006]. However, interesting works providing efficient graph compression techniques for potentially huge graphs have been recently proposed [Washio and Motoda, 2003; Tian et al., 2008].

[6] http://less.aksw.org

Facet-oriented visualization. Faceted browsing [Yee et al., 2003] has been widely adopted for many RDF dataset, spanning from DBpedia to DBLP. In faceted browsing the information space is partitioned using the "facets", which represent the important characteristics of an information element. The goal of faceted browsing is to restrict the search space to a set of relevant resources, by selecting, manually or automatically, the most important facets and values. The facet theory can be directly mapped to navigation in RDF data: information elements are RDF subjects, facets are RDF predicates and the values are the RDF objects.

An implementation for faceted navigation of arbitrary RDF data is presented in [Oren et al., 2006], where important facets are identified by automatically ranking the predicates that best represent and most efficiently navigate the dataset.

In [Hahn et al., 2010], authors present a faceted browser for Wikipedia. The system enables users to ask complex queries against the Wikipedia knowledge, by exploiting the Wikipedia infoboxes (i.e., the set of most relevant facts of an article displayed as a table of attribute-value pairs in the article Wikipedia page).

In [Yitzhak et al., 2008], a hierarchically faceted search implementation is described. Here, the facet values that are shown to the user are selected not only on the bases of their relevance with respect to the specific query, but also on the bases of their general importance.

Marbles[7] retrieves information about resources of interest by querying Sindice[8]. Sindice [Tummarello et al., 2007] ranks resources (i.e., RDF triples) retrieved by SPARQL queries exploiting external ranking services (as Google popularity) and information related to hostnames, relevant statements, and the dimension of the information sources. Marbles improves the user experience by presenting the resources as property-value pairs in a table. Different colors are used to distinguish the sources of the retrieved information, which are presented as a list of URIs.

Sig.ma[9] (Semantic Information MAshup) [Tummarello et al., 2010] retrieves and integrates Linked Data, starting from a single URI, by querying the Semantic Web and applying machine learning to the data found. Results are presented as a reorderable list of verified resources and links to potentially relevant information on the query subject; users may confirm or reject the relevance of each resource.

[7] http://www5.wiwiss.fu-berlin.de/marbles

[8] http://sindice.com

[9] http://sig.ma

Finally, in [Aleman-Meza et al., 2003], the context of entities in a RDF graphs is analyzed and exploited to find and present to the user the top relevant associations and information.

Faceted browsing improves usability over current interfaces and RDF visualizers as it provides a better information lookup with respect to keyword searches. Nevertheless, faceted interfaces are domain-dependent, do not allow to navigate through relations different from the ones explicitly represented in the dataset, and they become difficult to use for the users as the number of presented facets grows.

2.3 Contributions with respect to the state of the art

With respect to the state of the art, and with respect to the requirements presented in Chapter 1, the main contribution of the Ph.D. thesis is to provide a semantic clouding approach that is able to span across multiple Webs, dealing with different kinds of web resources. In fact, all of the aggregation and organization techniques and approaches described in this Chapter are focused only on a specific kind of web resource (e.g., web pages, news, Semantic Web resources). As far as we know, our semantic clouding approach represents a first attempt to bridge the gap between Semantic Web resources (typically managed in Linked Data) and other kinds of web resources, such as, for example, tagged and microdata resources. As a consequence, such approach also requires new kinds of matching techniques, which are able to compare heterogeneous web resources by calculating their semantic similarity.

The contributions of the Ph.D. thesis can be summarized as follows.

- Definition of a *cross-web* approach considering the different kinds of available web resources (e.g., tagged resources, microdata resources, Semantic Web resources), and considering both objective and subjective information.
- Definition of *in-cloud* as a new intuitive data structure for organizing relevant web resources for a given target entity on the basis of their prominence and closeness, capturing both their relevance with respect to the target entity and their reciprocal level of similarity.

- Definition of *matching techniques* for comparing different kinds of web resources by considering the nature of information they represent. In particular, such techniques have been specifically tailored to data clouding purposes to enable a more effective and smart browsing of the *in*-cloud, rather than the syntax-based data linking techniques currently available.

Chapter 3

The proposed approach

In recent years, the traditional World Wide Web based on "user-consuming" applications and informative web pages has changed into a more complex vision composed of a plurality of Webs, where semantic-intensive applications [Alexander et al., 2009; Hausenblas, 2009] as well as interactive "user-generating" platforms like microblogging, and personal news feeds [Chi, 2008; Koutrika et al., 2009b] are becoming more and more popular. In particular, we can mainly distinguish three different kinds of Web: the *Social Web*, the *Semantic Web*, and the *Web of Documents*. Thus, different kinds of web resources are currently available, coming respectively from the different kinds of Web. The Social Web resources include, for example, posts and news published by users in social networks and microblogging systems, the Semantic Web resources include RDF/OWL descriptions, and the resources coming from the Web of Documents include the traditional web pages.

In this scenario, the research efforts towards the development of solutions for organizing this huge amount of web resources according to data clouding or similar approaches is still at an initial stage [Kuo et al., 2007; Hernández et al., 2008; Koutrika et al., 2009a]. We propose a semantic clouding approach which is able to consider and retrieve different kinds of web resources, coming from the Social Web, the Semantic Web, and the Web of Documents. In particular, our approach is motivated by the following. A user is typically interested in finding all the available information about a given target entity. Relevant information about the target entity can be found in each one of the Webs which are currently available. Objective information about the target

entity can be extracted from the Web of Documents, subjective information about the target entity (e.g., comments, opinions from different users) can be extracted from the Social Web, and relations of the target entity with other entities can be extracted from the Semantic Web. In particular, our semantic clouding approach analyzes and organizes the following web resources, coming respectively from the Web of Documents, the Social Web, and the Semantic Web.

- **Tagged resources.** For what concerns the Web of Documents, we consider the web pages which have been previously tagged, manually or automatically. To this end, for our experiments, we have considered the web pages included in the tagging systems, that is, the ones that have been tagged by users who visited such pages. A tagged resource is characterized by a set of plain tags expressed by the system users to describe its content. Tags are single words chosen arbitrarily by users. An annotated web page in a system like del.icio.us and a tagged picture stored in Flickr are examples of tagged resources.

- **Microdata resources.** For what concerns the Social Web resources, we specifically consider the so called "microdata", which are referred to the posts published by users in social networks and microblogging systems. A microdata resource is characterized by a short textual content and a set of metadata/properties, like title, author, and creation date, that are commonly employed to describe published items. A user post in a social network system like Twitter and a news published in a personal RSS/ATOM feed are examples of microdata resources. Further examples of microdata resources are the new upcoming standards for the web communication, like HTML5[1] and microformats[2]. In particular, HTML5 will provide a microdata vocabulary to associate a nested semantics with the specific contents of a web page[3]. The microdata tags of HTML5 will be acquired as properties of a microdata resource, and will be subsequently considered during the *in*-cloud construction.

- **Semantic Web resources.** They are extracted from RDF(S) knowledge repositories and OWL ontologies. A Semantic Web resource is a structured description

[1] http://www.w3.org/TR/html5

[2] http://microformats.org

[3] http://www.w3.org/TR/html5/microdata.html

of an individual and it is characterized by a set of assertions denoting its specification in the web document of origin. A RDF description and an OWL instance are examples of Semantic Web resources.

In order to provide to the user all the relevant information about the target entity, it is important to develop solutions which are able to consider, analyze, and retrieve all these kinds of web resources. In fact, as shown in Chapter 2, there are still no approaches which are able to consider different kinds of web resources at the same time.

The aim of our semantic clouding approach is to provide to the user who makes the query a visual answer in the form of a cloud, where all the retrieved web resources, coming from the different Webs, are organized on the basis of their relevance and on the basis of their respective similarity. Such goal is achieved through the notion of *in*-cloud. An *in*-cloud is a collection of different kinds of web resources which are relevant for a given target entity and it provides a *cross-web*, *disciplined*, and *intuitive* information organization structure. In an *in*-cloud, all the retrieved web resources are properly arranged for agile, similarity-driven consultation by the user. In particular, the web resources included in an *in*-cloud are selected because of their *closeness* to the target entity, where by "closeness" we mean that we collect together not only those resources that represent different descriptions of the target entity but also those other resources that are similar or connected to the target entity. According to this approach, if the target entity of interest is for example a movie, an *in*-cloud for this movie will collect together different descriptions of the movie, descriptions of other movies that are similar to the target, and information about the movie cast, characters, production, and so on. Moreover, a measure of *relevance* is associated, also in a visual way, to each retrieve web resource, in order to show the "importance" of the corresponding web resource with respect to the target entity. A formal definition of *in*-cloud will be given in Chapter 5. The main featuring properties of *in*-clouds are the following.

- **Cross-webness.** Web resources in an *in*-cloud come from multiple Webs to provide a comprehensive picture of all available information, both *objective* and *subjective*. In fact, an *in*-cloud pervasively collects together objective information produced by conventional web data resources and subjective information derived from Social Web resources. In such a way, the official information about the target entity that is usually provided by web sites and broadcasters is com-

plemented with the so-called user generated content as it can be derived from microblogging and other similar kinds of information sources. Coming back to our movie *in*-cloud example, data about a movie that can be retrieved from the Semantic Web, including title, plot, duration, characters, are combined together with users reviews, opinions, comments that can be derived from the Social Web.

- **"Discipliness".** Web resources in an *in*-cloud are not only those directly related to target entity (i.e., those trivially matching the target entity) but also those that are in some way related to the target and are close to it. To properly highlight the different levels of closeness, the prominence of web resources is made explicit in the *in*-cloud in a way that the user can explore the *in*-cloud from the more prominent resources and then browse through the *in*-cloud following closeness paths. Due to its organization, an *in*-cloud not only fulfills the user needs but also anticipates them in a sense, in that the most prominent resources are complemented with supplementary information giving the overall picture of the target actually available across the Webs.

- **Intuitiveness.** The *in*-cloud information organization borrows the graphical representation commonly used for folksonomies and tag clouds that become popular on the Social Web. In particular, the graph-based organization of *in*-clouds makes explicit not only the prominence of each web resource with respect to the target entity but also the closeness level of web resources with one another. This supports the user in browsing the *in*-cloud more effectively by prominence and closeness of web resources here included. The *in*-clouds can be exploited by an agent, either a final user or a software agent like a search engine, to collect an overall picture of the available knowledge about a given target entity, rather than to retrieve a specific detail. This potentially leads to several possible applications consuming web data, ranging for example from a new generation of news aggregators to on-demand mashup applications.

An example of *in*-cloud is shown in Figure 3.1, collecting web resources related to the target entity "Star Wars". The nodes in the *in*-cloud represent web resources, while the edges between nodes denote the closeness degree holding between the corresponding web resources. The dimension of each node is proportional to the prominence of the corresponding web resource for the target entity "Star Wars". We can observe that web resources in the *in*-cloud are not only those directly related to this popular movie,

such as the titles of the six movies of the Star Wars saga, but also resources that are close to the movie saga even if not directly matching the target, such as some of the most important characters in the movies.

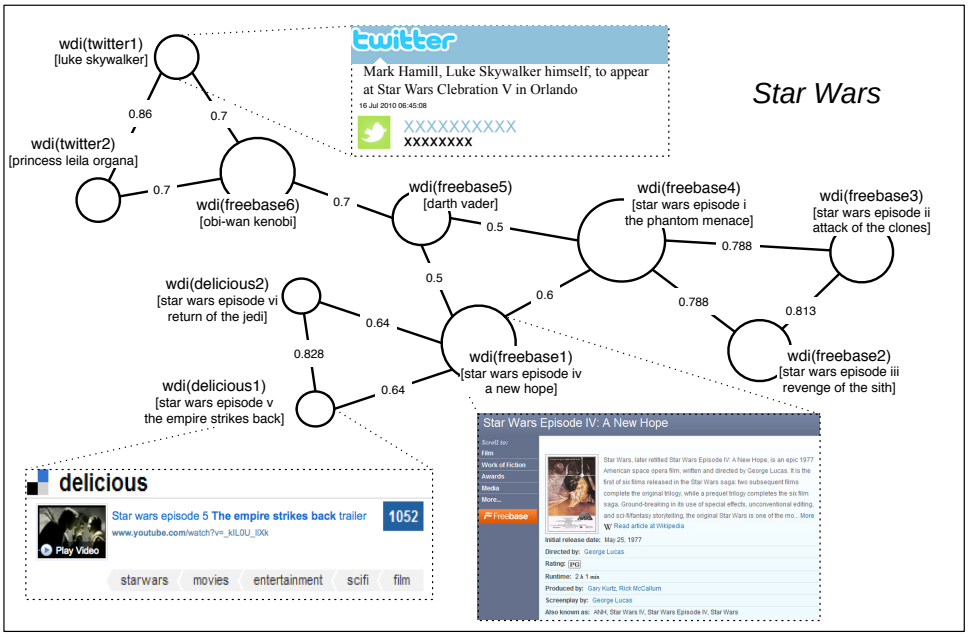


Figure 3.1: Example of *in*-cloud for the entity “Star Wars”

3.1 The proposed semantic clouding approach

In Figure 3.2, we show the semantic clouding approach developed for *in*-cloud construction. The approach is articulated in three phases: i) modeling of web resources, ii) classification of web resources, and iii) clouding of web resources.

The first step is to model all the web resources we want to consider in terms of “web data items”, according to a unified model that we introduce, called WDI (Web Data Item) model. Then, the classification of web resources aims at grouping together web resources having a high level of closeness, that is evaluated by exploiting matching techniques. Finally, the clouding of web resources is based on the results of the classification activity and aims at constructing the appropriate *in*-cloud organization

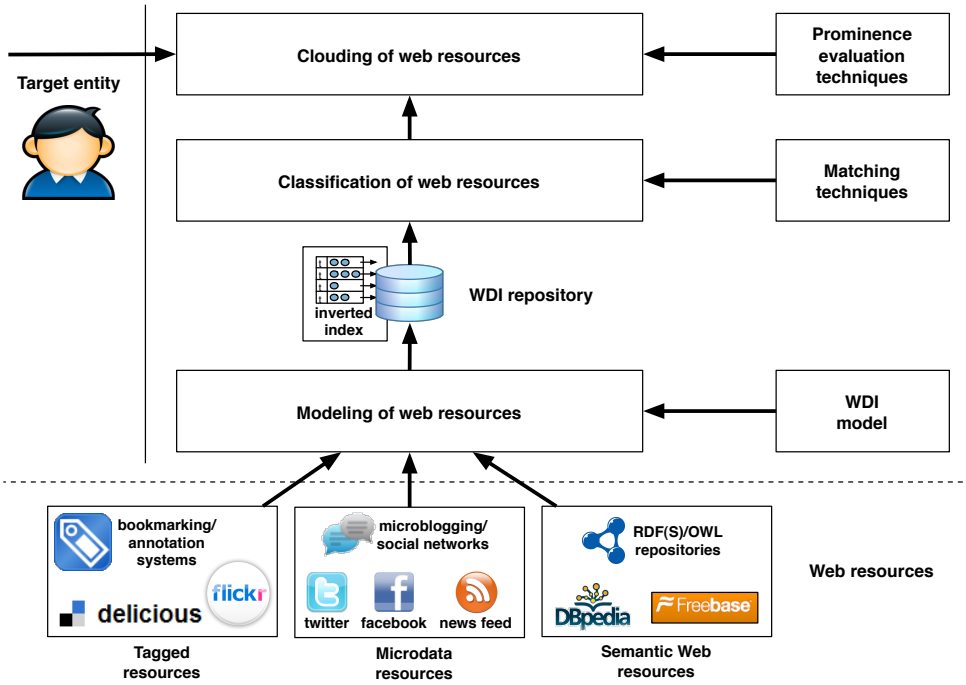


Figure 3.2: The semantic clouding approach

for a given target entity by prominence and closeness levels. The prominence of web resources for the target entity is calculated by exploiting suitable prominence evaluation techniques.

3.2 Running example

As a reference running example we consider the *in*-cloud construction for the target entity "Star Wars" (see Figure 3.1) by exploiting web resources coming from del.icio.us[4], Twitter[5], Freebase[6], and BDpedia[7] to stress the cross-webness property. To extract data from these web sources, we developed focused acquisition tools in the framework of our clouding prototype. In particular, the running example involves the following

[4] http://www.delicious.com
[5] http://twitter.com
[6] http://www.freebase.com
[7] http://dbpedia.org

acquisition tools.

- *Acquisition tool for social annotation systems*. This tool allows to exploit the dataset described in [Wetzker et al., 2010] that contains about 142 million of del.icio.us bookmarks, by also supporting the (optional) submission of keyword-based queries to enforce a selective resource acquisition.

- *Acquisition tool for RSS feed and Twitter*. This tool allows to specify a set of RSS channels to acquire and it supports post extraction from the search service of Twitter[8] through the Twitter API. Selective resource acquisition is also enforced by specifying conditions based on keywords of interest and range of dates that posts have to satisfy for being acquired.

- *Acquisition tool for ontologies and Linked Data*. This tool allows to acquire instances from OWL files and to extract Linked Data resources from Freebase and DBpedia repositories. For Linked Data acquisition, two kinds of filtering operations are enforced. Filtering based on a seed of interest, which acquires all the Linked Data resources concerning with a specific URI given as input (i.e., the seed); and filtering based on keywords, which acquires all the Linked Data resources that contains at least one of the specified keywords.

3.3 The WDI (Web Data Item) model

To build *in*-clouds by mixing up both objective and subjective information about a certain target entity requires the capability to deal with a variety of web resources coming from the different Webs. As specified, we consider, in our semantic clouding approach, the tagged resources, the microdata resources, and the Semantic Web resources, extracted respectively from the Web of Documents, the Social Web, and the Semantic Web. Thus, we need to introduce a reference data model that is able to represent in a unified way all such different kinds of web resources, in order to calculate their respective level of similarity by comparing uniform representations. Since no other reference data models are currently available for representing in a uniform way different kinds of web resources, we define, for our semantic clouding purposes, the *WDI model*. The WDI model is based on the notion of *web data item* (wdi) to represent the metadata

[8] http://search.twitter.com

featuring the various kinds of web resources. Given a generic web resource *wr*, we define its wdi representation as a tuple of the form:

$$wdi(wr) = \langle \mathcal{T}, \mathcal{S}, \mathcal{L}, prov \rangle$$

where

- $\mathcal{T}(wr) = \{(t_1, f_1), \ldots, (t_n, f_n)\}$ is the *term equipment* of *wr*. $\mathcal{T}(wr)$ is defined as a set of pairs (t_i, f_i), with $i \in [1,n]$, where t_i is a term appearing in the specification of *wr*, like a concept name, a property name, a URI label or a literal, and f_i is the corresponding frequency (i.e., number of occurrences) of t_i in *wr*.

- $\mathcal{S}(wr) = \left\{(p_1^1, p_1^n, v_1, l_1), \ldots, (p_m^1, p_m^n, v_m, l_m)\right\}$ is the *structure equipment* of *wr*. $\mathcal{S}(wr)$ summarizes the structure of the resource *wr* and it is defined as a set of tuples (p_j^1, p_j^n, v_j, l_j), with $j \in [1,m]$, each one describing a property p_j of *wr*. A tuple of $\mathcal{S}(wr)$ allows to represent not only a conventional property with a name and a corresponding literal value, but also a property that represents a path of references in the specification of *wr*. This frequently occurs for example in RDF/OWL instances where the value of a property can be a reference to another property and the literal value appears after a path of property references. In particular, given a path of property references $p_j^1 \rightarrow \ldots \rightarrow p_j^n \rightarrow v_j$, p_j^1 denotes the name of the first property in the path, p_j^n denotes the name of the last property in the path (which coincides with p_j^1 in the case of conventional properties), v_j denotes the literal value of the last property p_j^n, and l_j denotes the length of the path from p_j^1 to p_j^n. This wdi representation of properties is motivated by two considerations. First, for property paths in a RDF/OWL resource with a length $n = 2$, a tuple of $\mathcal{S}(wr)$ is capable of representing the full path. Second, for property paths with $n \geq 3$, the idea to consider only p^1 and p^n is an effective trade-off between, on one hand, the need to provide a fixed-length representation for property paths, and, on the other hand, the need to capture the meaning of the property in the wdi representation.

- $\mathcal{L}(wr) = \{type_1, \ldots, type_l\}$ is the *logics equipment* of *wr*. $\mathcal{L}(wr)$ denotes the knowledge about *wr* derived through a reasoning process based on a model *O*, such as a RDF Schema or an OWL ontology. $\mathcal{L}(wr)$ is defined as the set of the types/classes $type_k$ of *O*, with $k \in [1,l]$, for which *wr* is recognized to be a valid instance.

- *prov* is the provenance of the web resource *wr* and it is expressed by the URI from which the resource has been acquired.

3.3.1 Representing web resources through the WDI model

Each web resource, either a tagged resource, a microdata resource, or a Semantic Web resource, is associated with its wdi representation as follows.

Tagged resources. Given a tagged resource tr, its wdi representation is only characterized by the term equipment $\mathcal{T}(tr)$, built on the basis of the annotations of tr, that is:

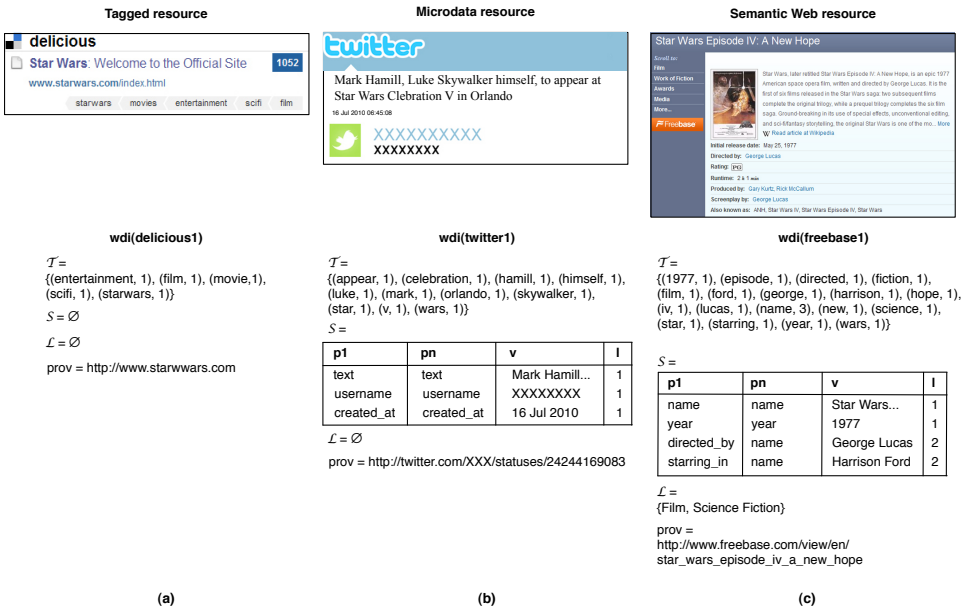
$$wdi(tr) = \langle \mathcal{T}; \emptyset; \emptyset; URI \rangle$$

A pair $(t_i, 1)$ is inserted in $\mathcal{T}(tr)$ for each annotation t_i, with $i \in [1, n]$, associated with the tagged resource tr. Before insertion in $\mathcal{T}(tr)$, an annotation is submitted to a normalization procedure for word-lemma extraction and for compound-term tokenization [Sorrentino et al., 2009]. Since neither structure nor logics information about tr can be derived from the annotations, we have that $\mathcal{S}(tr) = \emptyset$ and $\mathcal{L}(tr) = \emptyset$.

Running example. In Figure 3.3(a), we show an example of tagged resource taken from the del.icio.us annotation system. The considered tagged resource *delicious*1 is the official web site of the Star Wars movie which is annotated in del.icio.us with the tags "starwars", "movies", "entertainment", "scifi", and "film". The corresponding wdi representation is the following.

$$wdi(delicious1) = \langle \{(entertainment, 1), (film, 1), (movie, 1), (scifi, 1), (starwars, 1)\}; \emptyset; \emptyset; \texttt{http://www.starwars.com} \rangle$$

We note that the annotation tag "movies" is normalized into "movie" before insertion in $\mathcal{T}(delicious1)$. We also note that the annotation tag "starwars" is left in this form since it is not recognized as a compound term by conventional tokenization techniques. Advanced techniques for compound-term discovery based on lexical analysis can be exploited for this kind of annotation tags. Such techniques have been developed and described in [Varese and Castano, 2011], and they will be presented in Chapter 4.



$\mathcal{T}$ =
{(entertainment, 1), (film, 1), (movie,1), (scifi, 1), (starwars, 1)}
$\mathcal{S}$ = ∅
$\mathcal{L}$ = ∅
prov = http://www.starwwars.com

$\mathcal{T}$ =
{(appear, 1), (celebration, 1), (hamill, 1), (himself, 1), (luke, 1), (mark, 1), (orlando, 1), (skywalker, 1), (star, 1), (v, 1), (wars, 1)}
$\mathcal{S}$ =

p1	pn	v	l
text	text	Mark Hamill...	1
username	username	XXXXXXXX	1
created_at	created_at	16 Jul 2010	1

$\mathcal{L}$ = ∅
prov = http://twitter.com/XXX/statuses/24244169083

$\mathcal{T}$ =
{(1977, 1), (episode, 1), (directed, 1), (fiction, 1), (film, 1), (ford, 1), (george, 1), (harrison, 1), (hope, 1), (iv, 1), (lucas, 1), (name, 3), (new, 1), (science, 1), (star, 1), (starring, 1), (year, 1), (wars, 1)}

$\mathcal{S}$ =

p1	pn	v	l
name	name	Star Wars...	1
year	year	1977	1
directed_by	name	George Lucas	2
starring_in	name	Harrison Ford	2

$\mathcal{L}$ =
{Film, Science Fiction}
prov =
http://www.freebase.com/view/en/star_wars_episode_iv_a_new_hope

Figure 3.3: Examples of wdi representation for a) Tagged resource, b) Microdata resource, and c) Semantic Web resource

Microdata resources. Given a microdata resource mr, its wdi representation is characterized by a term equipment $\mathcal{T}(mr)$ and by a structure equipment $\mathcal{S}(mr)$, while $\mathcal{L}(mr) = \emptyset$, that is:

$$wdi(mr) = \langle \mathcal{T}; \mathcal{S}; \emptyset; URI \rangle$$

The term equipment $\mathcal{T}(mr)$ is built by extracting a set of featuring terms from the textual content of mr. Conventional text analysis techniques are applied to the content of mr with the goal of removing stop-words (e.g., articles, conjunctions, prepositions) and other special characters/symbols that are commonly employed in microblogging and news publishing systems (e.g., #, @ for Twitter posts). After text analysis, a pair (t_i, f_i), with $i \in [1,n]$, is inserted in $\mathcal{T}(mr)$ for each term t_i of the mr content, where f_i is the frequency of t_i in the content of mr. Moreover, state of the art techniques for keyphrase extraction can be suitably employed to populate $\mathcal{T}(mr)$ with the most featuring phrases of a web resource (see for example [Chen et al., 2005]). In this case, each detected keyphrase is inserted in $\mathcal{T}(mr)$ as an atomic element and the corresponding frequency is set to 1. The structure equipment $\mathcal{S}(mr)$ is built on the properties in the description of mr. A microdata resource is characterized by a flat structure of conven-

tional properties (p_j, v_j), where p_j is a property name and v_j is a corresponding literal value like a string or an integer. For each p_j of mr, with $j \in [1,m]$, a tuple $(p_j, p_j, v_j, 1)$ is defined in $\mathcal{S}(mr)$.

Running example. In Figure 3.3(b), an example of Twitter microdata resource and its corresponding wdi representation is provided. In this case, the term equipment $\mathcal{T}(twitter1)$ of the wdi representation of the microdata resource $twitter1$ is derived from the post content. Moreover, the structure equipment $\mathcal{S}(twitter1)$ is composed by the properties *text*, *username*, and *created_at* that characterize a Twitter post. The value of the *text* property is the post content, while the value of *username* and *created_at* are the identifier of the post author and the creation date of the post, respectively.

Semantic Web resources. Given a Semantic Web resource *sr*, its wdi representation is defined as follows:

$$wdi(sr) = \langle \mathcal{T}; \mathcal{S}; \mathcal{L}; URI \rangle$$

where the various equipments are built by considering the specification of *sr*, namely the RDF graph $\mathcal{G}^{sr}$ that can be derived from the model O (i.e., RDF(S)/OWL resource) associated with *sr*. The term equipment $\mathcal{T}(sr)$ is defined as the set of all the terms appearing in the nodes and edges of the graph $\mathcal{G}^{sr}$, such as concept names, property names, URI labels, comments, and literals. A pair (t_i, f_i), with $i \in [1,n]$, is inserted in $\mathcal{T}(sr)$ for each term t_i and associated frequency f_i in $\mathcal{G}^{sr}$. Also in this case, procedures for term normalization, word-lemma extraction, and compound-term tokenization are executed on t_i before insertion in $\mathcal{T}(sr)$. The structure equipment $\mathcal{S}(sr)$ is built by adding a new tuple (p_j^1, p_j^n, v_j, l_j) for each path $j \in [1,m]$ of property references $p_j^1 \rightarrow \ldots \rightarrow p_j^n \rightarrow v_j$ with length l_j in $\mathcal{G}^{sr}$ between the *uri* of the resource *sr* and a literal value v_j. The logics equipment $\mathcal{L}(sr)$ is built through a reasoning process based on the model O. In particular, a type name $type_k$, with $k \in [1,l]$, is inserted in $\mathcal{L}(sr)$ denoting the fact that *sr* is a valid instance of $type_k$ in O.

Running example. In Figure 3.3(c), we show an example of the Semantic Web resource $freebase1$ taken from Freebase describing the movie Star Wars Episode IV (title "A New Hope"). The RDF graph $\mathcal{G}^{freebase1}$ of $freebase1$ is shown in Figure 3.4.

The term equipment $\mathcal{T}(freebase1)$ is composed by the terms appearing in the property names and literals of $\mathcal{G}^{freebase1}$. Starting from $URI01$, which denotes the URI of

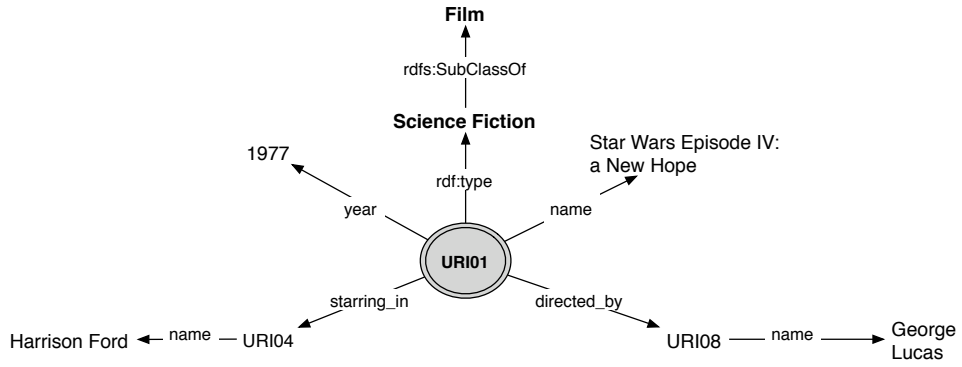


Figure 3.4: The RDF graph for the Semantic Web resource $freebase1$ (description of Star Wars Episode IV)

$freebase1$, we can build the structure equipment $\mathcal{S}(freebase1)$ by exploiting the various paths in $\mathcal{G}^{freebase1}$. We recognize two paths of length 1 associated with the properties *name* and *year* of $URI01$, for which the first two elements of $\mathcal{S}(freebase1)$ are inserted. The last two elements of $\mathcal{S}(freebase1)$ correspond to the properties *starring_in* and *directed_by* of $URI01$, respectively. In these two cases, the path length is set to 2 since the instances $URI04$ and $URI08$ need to be traversed to reach a literal value. The property name and the property filler of $URI04$ and $URI08$ are used to populate the parameters p^n and v of these two elements of $\mathcal{S}(freebase1)$, respectively. The logics equipment $\mathcal{L}(freebase1) = \{Film, Science\ Fiction\}$ is determined by exploiting the types of $URI01$ in $\mathcal{G}^{freebase1}$, including also those inherited.

3.3.2 The WDI repository

The web data items (i.e., the wdi representations) denoting the web resources acquired from the different Webs, are stored in a support repository, called *WDI repository*. In such a repository, an inverted indexing structure [Baeza-Yates and Ribeiro-Neto, 1999] is defined to enforce an efficient retrieval of the corresponding web resources of interest. The index terms composing the inverted structure are derived from the term equipments $\mathcal{T}(wr)$ for each resource description in the WDI repository. The use of the inverted index structure for the construction of *in*-clouds will be described in Chapter 5.

Chapter 4

Matching techniques for *in*-cloud construction

Matching techniques are required in our approach in order to find the web resources that are referred to the same real-world object or, more in general, that are somehow similar. To this end, we have studied the state of the art instance matching techniques, and we have then developed a set of matching techniques specifically thought to compare web resources which are mainly featured by a textual description, such in the case of tagged resources. Thus, different kinds of matching techniques have been developed [Castano et al., 2009b,c; Ferrara et al., 2009, 2010; Montanelli et al., 2010; Varese and Castano, 2011], in order to efficiently analyze and exploit the textual description of each type of web resource. Moreover, term matching techniques assume a central role in the effective classification of heterogeneous kinds of web resources. In fact, as instance matching techniques are specifically thought to compare pairs of ontology instances (i.e., Semantic Web resources), term matching techniques can be used to compare not only pairs of tagged resources or Social Web resources, but also to compare tagged and Social Web resources, which are mostly characterized by a short textual description, with structured web resources, such as Semantic Web resources.

In this chapter, we first present a literature survey of instance matching and record linkage techniques (see Section 4.1). Then, we describe the matching techniques that we have developed for semantic data clouding (see Section 4.2).

4.1 Instance matching

The same real-world object can be described multiple times in different knowledge repositories, possibly using different perspectives and by emphasizing different properties of interest. In fact, every real-world object (e.g., a person, a place, an event) can appear on the web within a number of different documents with heterogeneous representations called *instances* or *individuals*. The capability of finding similar object descriptions assumes particular relevance in the field of Semantic Web, to promote effective resource sharing on the global scale and to correctly interoperate/reuse individual knowledge chunks coming from disparate information repositories, disregarding their specific URIs. Such task is called *instance matching*, and consists in finding instances (i.e., object descriptions), coming from different sources (e.g., OWL ABoxes, RDF descriptions), which describe the same real-world object in a different and heterogeneous way. Formally, the instance matching problem can be defined as follows.

Given two instances i_1 and i_2 as input, instance matching is defined as the process of comparing i_1 and i_2, in order to produce as output a value $v \in \{0,1\}$. If $v = 0$, it means that i_1 and i_2 describe different real-world objects. Otherwise (i.e., if $v = 1$), it means that i_1 and i_2 are deferred to the same real-world object.

Approaches and techniques for instance matching are currently employed in a number of application fields. For example, in the Semantic Web, instance matching is exploited to address the so-called *identity recognition* problem. In this field, instance matching has the goal to support discovery and reuse on the web of a unique identifier for the set of instance descriptions that is recognized as referring to the same real-world object. Some contributions in this direction have been focused on defining techniques and approaches for generation and management of identifiers at object-level, like, for example, the OKKAM project [Bouquet et al., 2006, 2008]. Other approaches have been proposed for the unification of different URIs associated to the same object [Nikolov et al., 2008]. In the field of semantic integration, instance matching can be used to determine the set of matching concepts to integrate in two considered knowledge sources. To this end, the similarity between two concepts is evaluated by measuring the "significance" in the overlap of their respective instance sets, and two instances are considered as overlapping according to their level of similarity [Wang

et al., 2006; Isaac et al., 2007]. Moreover, instance matching is currently demanded in the field of ontology management where it is invoked to support domain experts in performing ontology changes through advanced, and possibly automated, techniques. For example, instance matching is used to correctly perform the insertion of new instances in a given ontology (i.e., *ontology population*) and to discover the possible similarity mappings between a new incoming instance and the set of instances already represented in the ontology. Mappings among instances can be exploited to enforce a query answering mechanism based on instance similarity. More recently, some new techniques have been proposed to specifically match ontology instances [Isaac et al., 2007] and to identify similar web resources [Madhavan et al., 2007; Langegger et al., 2008].

Up to now, techniques for instance matching are mostly borrowed from those developed for *record linkage*, which has been widely studied in the databases community [Fellegi and Sunter, 1969; Newcombe, 1988; Hernández and Stolfo, 1995; Winkler, 1999]. In the database community, record linkage is defined as "the task of quickly and accurately identifying records corresponding to the same real-world entity from one or more data sources" [Gu et al., 2003]. As this problem is very general, in the literature, it is known under different names (e.g., data deduplication, duplicate detection, merge/purge problem), according to the specific requirements that need to be satisfied and to the goals that need to be pursued [Wang et al., 2006; Zhou and Hansen, 2006; Yan et al., 2007].

In the following, we will focus on the problem of instance matching by classifying existing approaches proposed for record linkage and by explaining how they can be/are being used for instance matching purposes. For a survey of the *schema matching* approaches, see [Rahm and Bernstein, 2001] and [Shvaiko and Euzenat, 2005].

Main approaches for record linkage were initially proposed for database applications, as a solution for deduplication. In particular, given a set of records $r_1, \ldots, r_n$ as input (i.e., the tuples belonging to one or more database relations), the deduplication process consists in firstly detecting different records referring to the same real-world object (duplicates or matching records), and secondly in removing duplicates through appropriate record merge/unification operations. For our purpose, we will focus on

record linkage techniques for duplicate detection, since they can be adapted to work on instance matching.

As shown in Figure 4.1, these techniques can be classified into two different categories, corresponding to two different levels of granularity: the *value-oriented techniques* and *record-oriented techniques*.

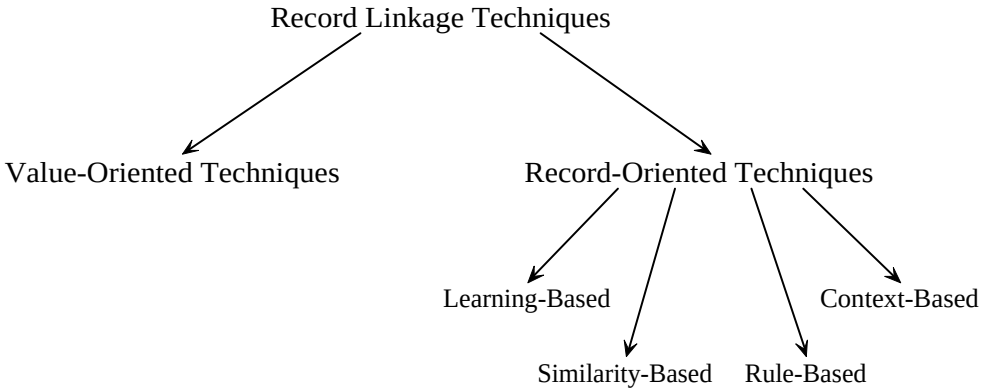


Figure 4.1: A basic classification of existing record linkage techniques

For record linkage, a record r_i is represented as a vector $\overline{r_i} = [v_1, \ldots, v_m]$, where m is the number of its featuring attributes and v_j is the value of the j-th attribute. Given a pair of records r_1 and r_2, the goal of value-oriented techniques is to determine the similarity $sim(v_h, v_k)$ of values v_h and v_k, where $v_h \in \overline{r_1}$ and $v_k \in \overline{r_2}$, for each pair of corresponding attributes of r_1 and r_2. Record-oriented techniques aim at computing the overall similarity $sim(r_1, r_2)$ of r_1 and r_2, in order to determine whether r_1 and r_2 refer to the same real-world entity.

Value-oriented techniques. These techniques work at the value granularity under the assumption that the similarity level of two records r_1 and r_2 can be derived by matching the values of their comparable attributes. For each specific datatype attribute, appropriate matching techniques are provided to calculate the similarity of attribute values. As an example, approaches for matching numerical values use conversion functions to determine how to transform values of a source datatype (e.g., real

values) into corresponding values of a target datatype (e.g., integer values). However, most of the work on value-oriented matching has been focused on computing similarity of string attributes due to the fact that string data are the most frequently used datatype in database and knowledge repositories for real-world entity descriptions. Different techniques have been developed in order to manage specific kinds of errors/differences within string values. *Character-based techniques*, like the Edit Distance, the Smith-Waterman Distance, and the Jaro Distance, are specifically suited for comparing string values and recognize typographical errors (e.g., "Coputer Science", "Computer Sceince"). They basically compute the number of common characters of two strings. *Token-based techniques*, like the Cosine Similarity, TF-IDF, and the Q-Gram distance, are able to manage the use of different conventions for describing data (e.g., "John Smith", "Smith, John"). In this case, the similarity of two strings is calculated by analyzing their common patterns (tokens). Finally, *phonetic-based techniques*, like Soundex, NYSIIS, and Metaphone, try to measure the phonetic similarity of different strings, even if their textual representation is very different (e.g., "Kageonne", "Cajun"). These techniques analyze the position of consonants and vowels.

Such techniques are currently used by all the instance matching tools to perform string matching operations.

Record-oriented techniques. When the similarity value $sim(v_h, v_k)$ of each pair of corresponding attribute values of two considered records r_1 and r_2 has been calculated, it is possible to decide if, given a threshold, r_1 and r_2 can be classified as matching or non-matching records. The set of similarity values of single pairs of attribute values is then given as input to a decision engine, whose aim is to classify r_1 and r_2 as matching or non-matching records, by analyzing them as a whole. The decision engine works under the rules of a certain methodology, which in turn uses different techniques to compare and classify records. Such techniques can be classified in four categories: the learning-based techniques, the similarity-based techniques, the rule-based techniques, and the context-based techniques.

Learning-based techniques make use of a classifier in order to understand if two records refer to the same real-word entity or not. Thus, the classifier takes as input a set of instance pairs, together with the expected classification (i.e., matching or non-

matching records). If the training set is adequate, the system will then be able to correctly classify new input data. The main concern using these techniques is the need to find a good training data set. In fact, the training input has to cover all the possible situations but, at the same time, it has to be general enough to make the system able to discover the correct classification functions. This is a non-trivial task and it usually requires a manual selection. An example of tool using learning-based techniques is presented in [Singla and Domingos, 2004]. An example of *supervised* learning technique is used by ALIAS [Sarawagi and Bhamidipaty, 2002], which automatically classify record pairs that clearly refer to the same real-world entity as well as record pairs that clearly denote different real-world entities, and automatically selects ambiguous record pairs, which instead have to be classified by humans. Examples of approaches using *unsupervised* learning techniques (i.e., techniques which do not require the human intervention) are presented in [Verykios et al., 2000] and in [Christen, 2007, 2008a]. An alternative idea is to put already-classified data together with non-classified data, in order to reduce the amount of training information needed, still having good quality results. These methods are called *semi-supervised* learning techniques. An example of them is presented in [Pasula et al., 2002].

Learning-based techniques are being recently proposed also in the field of instance matching. For example, in [Wang et al., 2006], the authors propose to determine the set of matching instances stored in two considered ontologies by combining the results of different string matching functions (e.g., edit distance, cosine similarity) with a machine learning approach based on a SVM (Support Vector Machine) classifier. Different string matching functions are separately exploited to compare the values of the instance properties and to calculate their own set of mappings denoting the pairs of matching instances. The SVM classifier is then invoked to determine the final set of matching instances by considering the various sets of (potentially different) mappings computed by the string matching functions. A set of matching instances calculated on a reference domain ontology is used as training set for the SVM classifier.

Similarity-based techniques consider the input records as long attribute values. In this case, it is possible to use the same methods used to compare attribute values, such as string matching functions. Another approach to measure the similarity degree between two records is to calculate the average similarity of each pair of their attribute values [Dey et al., 1998]. If some information about the relative importance of each

attribute is available, the similarity of a record pair can be measured by calculating the weighted average of the similarity of each single pair of attribute values. The weight of each attribute can be manually specified by a domain expert [Dey et al., 2002] or it can be automatically determined through statistical analysis [Guha et al., 2004]. Finally, a further refinement of the instance matching process is to take into account the frequency each value occurs [Winkler, 2000]. In particular, a pair of matching attribute values will receive a high weight if these values occur with a low frequency within the domain, while they will receive a low weight otherwise. The idea is that records sharing a rare attribute value are more likely to refer to the same real-world entity. The main drawback of similarity-based techniques is the identification of a right threshold, in a way that distinguishing matching from non-matching records is reasonable. For example, the problem is to decide if two records having a similarity measure of 0.5 have to be considered as matching or not.

Rule-based techniques can be considered as a special case of similarity-based techniques. In fact, like similarity-based techniques, they assign a similarity value to each record pair but, differently from similarity-based techniques, they just produce a boolean output, namely 1 if the input records refer to the same real-world entity, and 0 otherwise. The idea behind these techniques is that, even if a key attribute is not available, it is still possible to identify a set of attributes that collectively are able to univocally distinguish each record [Wang and Madnick, 1989]. This attribute set is usually determined by domain experts [Hernández and Stolfo, 1998] and it can thus be exploited to identify heuristic rules which can help to find records referring to the same real-world entity. For example, if two records denoting persons share the same value on attributes *surname* and *address*, there is a very high probability that the considered records refer to the same person. Rule-based techniques produce very precise matching results, but they have the drawback that they are domain-dependent and that it can be difficult to find good heuristic rules for the considered domain.

Context-based techniques are generally based on the idea of performing record matching by considering not only their attribute values, but also their relationships with other records. In other words, records connected with the input records are considered to constitute their context. Thus, given two records, their similarity is computed by considering also the similarity value of each pair of records in their context. An exam-

ple of these techniques is presented in [Singla and Domingos, 2004]. Unlike classical methods based on the independent comparison of record pairs, this work proposes to analyze the records from one or more sources all together, by considering their shared attribute values. In particular, the process of finding duplicates is represented as an undirected graph where records sharing the same attribute values are linked together. Another example is presented in [Bhattacharya and Getoor, 2004]. In this work, the records to analyze are first clustered, and then, all the records within the same cluster are matched, in order to find duplicates. The deduplication process is iterative because matching records are linked together and, as new duplicates are discovered, the distance between clusters is updated, potentially leading to the discovery of new duplicates.

4.2 Matching techniques for closeness evaluation

Taking into consideration the presented instance matching techniques, we have developed a set of matching techniques for evaluating the level of closeness (i.e., similarity) between the web data items stored in the WDI repository. For data clouding, the choice of the matching techniques to use has to comply with the nature and the different complexity that can characterize the different web resources, and consequently, their corresponding wdi representations. For example, when matching is invoked for comparing tagged resources, we need to consider that the closeness evaluation can be only based on term equipments. Structure and logics equipments can be additionally exploited when matching the web data items of microdata and Semantic Web resources, respectively. Moreover, the techniques for matching web data items have to cope with the fact that the closeness evaluation can involve heterogeneous web data items. Such different situations are summarized in Table 4.1.

	Tagged resources	**Microdata resources**	**Semantic Web resources**
Tagged resources	$\mathcal{T}$	$\mathcal{T}$	$\mathcal{T}$
Microdata resources	$\mathcal{T}$	$\mathcal{T}, \mathcal{S}$	$\mathcal{T}, \mathcal{S}$
Semantic Web resources	$\mathcal{T}$	$\mathcal{T}, \mathcal{S}$	$\mathcal{T}, \mathcal{S}, \mathcal{L}$

Table 4.1: Matching of the web data items

For example, if we match a tagged resource wr_i against a microdata resource wr_j, matching can be executed between a web data item (i.e., $wdi(wr_i)$) characterized only

by the term equipment and a web data item (i.e., $wdi(wr_j)$) with both term and structure equipments, respectively. We note that the term equipment is the only equipment always defined in the wdi representation of all the web resources. Moreover, for a tagged or a microdata resource, the term equipment captures most of the informative content of the whole web resource. For this reason, term matching techniques play a crucial role for closeness evaluation, and thus, in developing our matching techniques, we specifically focused on providing flexible similarity functions which are able to consider the different kinds of similarity (i.e., syntactic, semantic, terminological, and linguistic) holding between terms. Furthermore, term matching techniques are also exploited by more articulated matching techniques to evaluate the structural similarity between web resources. Then, the term and the structural similarity coefficient of a given pair of web data items are combined, in order to compute their comprehensive closeness coefficient value, that is used in the hierarchical clustering procedure of Figure 5.1.

4.2.1 Term similarity

The term similarity coefficient $tsim(wdi(wr_i), wdi(wr_j)) \in [0,1]$ of two web data items $wdi(wr_i)$ and $wdi(wr_j)$ is proportional to the number of matching terms in their corresponding term equipments $\mathcal{T}(wr_i)$ and $\mathcal{T}(wr_j)$, as follows:

$$tsim(wdi(wr_i), wdi(wr_j)) = \frac{2 \cdot |t_x \sim t_y|}{|\mathcal{T}(wr_i)| + |\mathcal{T}(wr_j)|}$$

We use $t_x \sim t_y$ to denote that the terms $t_x \in \mathcal{T}(wr_i)$ and $t_y \in \mathcal{T}(wr_j)$ are matching terms. To detect whether $t_x \sim t_y$, we rely on the similarity function $sim(t_x, t_y)$. In particular, we have that $t_x \sim t_y \iff sim(t_x, t_y) \geq t$, where $t \in (0,1]$ is a matching threshold denoting the minimum level of term similarity required to consider two terms as matching terms. The $sim(t_x, t_y)$ value is calculated as follows: where w_{syn}, w_{sem}, w_{ter}, and

$$\begin{aligned} sim(t_x, t_y) \quad = \quad & w_{syn} \cdot sim_{syntactic}(t_x, t_y) + \\ & w_{sem} \cdot sim_{semantic}(t_x, t_y) + \\ & w_{ter} \cdot sim_{terminological}(t_x, t_y) + \\ & w_{lin} \cdot sim_{linguistic}(t_x, t_y) \end{aligned}$$

w_{lin} are weights assigned to the syntactic similarity $sim_{syntactic}$, the semantic similarity

$sim_{semantic}$, the terminological similarity $sim_{terminological}$, and the linguistic similarity $sim_{linguistic}$, respectively, with $w_{syn} + w_{sem} + w_{ter} + w_{lin} = 1$. The weight associated with each kind of similarity can be set by the user according to the specific need. In particular, the different kinds of similarity can be analyzed in an independent or combined way. If the user decides to consider only one kind of similarity, the weight associated with the corresponding similarity function is set to 1, and remaining weights to zero. If the user decides to combine $n \in \{1,2,3,4\}$ different kinds of similarity, the weight associated with each corresponding similarity function is set to $1/n$, and remaining weights to zero.

The different similarity functions, namely the syntactic similarity, the semantic similarity, the terminological similarity, and the linguistic similarity, are described in the following. These functions have been conceived to fully exploit information provided by terms included in the term equipments for a flexible similarity evaluation based on different term characteristics. In particular, the syntactic similarity is calculated by using conventional string matching functions, and it is mainly suited to recognize syntactic variations of the same term, including for instance typographical errors, or similar terms belonging to different grammar categories. The semantic similarity determines the level of matching on the basis of the co-occurrence between terms. The terminological similarity analyzes and exploits the WordNet relations between terms. Finally, the linguistic similarity takes into account knowledge about compound words/abbreviations and their related terms.

Term matching techniques have been evaluated and applied to two real datasets (i.e., the PINTS Experiments Data Sets[1] [Görlitz et al., 2008]), containing tags crawled during 2006 and 2007 from two different tagging systems, namely del.icio.us[2] and Flickr[3]. Both such datasets consist in a collection of tag assignments. In particular, the del.icio.us dataset contains 634736 tags, 213428 resources, and 6234 users, while the Flickr dataset contains 1389350 tags, 380001 resources, and 16235 users. A more detailed description and evaluation of these term matching techniques are presented

[1] http://www.uni-koblenz-landau.de/koblenz/fb4/AGStaab/Research/DataSets/PINTSExperimentsDataSets/index_html

[2] http://del.icio.us

[3] http://www.flickr.com

in [Varese and Castano, 2011].

Syntactic similarity function. The syntactic similarity function analyzes the syntactic similarity of a pair of terms (t_x, t_y). To calculate such similarity, we used the open source SimMetrics library[4], which provides the most popular string matching functions, such as the Levenshtein Distance, the Cosine Similarity, the Jaccard Similarity, the Jaro Distance, the Q-Gram Distance [Navarro, 2001]. Formally, the syntactic similarity function is defined as follows.

$$sim_{syntactic}(t_x, t_y) = getSimilarity(t_x, t_y)$$

Where *getSimilarity* is the specific string matching function used for calculating the syntactic similarity of the pair of tags (t_x, t_y). For the evaluation, we used as default the Levenshtein Distance [Levenshtein, 1966], which has been selected because it works well in most situations occurring in the analyzed datasets. The Levenshtein Distance of a given pair of strings (s_i, s_j) is calculated as the minimum number of edits (i.e., insertions, deletions, substitutions of single characters) needed to transform s_i into s_j. In the *getSimilarity* function, the Levenshtein Distance of a given pair of tags (t_x, t_y) is normalized with the length of the longer tag among t_x and t_y, as follows.

$$getSimilarity(t_x, t_y) = 1 - \frac{LevenshteinDistance(t_x, t_y)}{max\left\{length(t_x), length(t_y)\right\}}$$

Where *LevenshteinDistance* is the function which calculates the Levenshtein Distance of the pair of tags (t_x, t_y), and $length(t_x)$ and $length(t_y)$ are the functions which calculate the length of t_x and t_y, respectively.

Example. The results of matching the term "technology" against the del.icio.us and Flickr datasets using the syntactic similarity function are shown in Figure 4.2. Using this kind of similarity, the resulting matching terms are syntactically similar to the target keyword. Thus, terms containing typographical errors (e.g., "technoloogy", "technologie"), or which are non-English words (e.g., "teknologi") are also returned as matching. These results can also be useful in that they are related to the term "technology" as well. However, some of the results can be misleading (e.g., "ethnology"), as they have nothing to do with "technology", even if they are syntactically similar

[4] http://www.dcs.shef.ac.uk/~sam/simmetrics.html

Keyword	Top-10 matching terms	Similarity value
technology	technoloogy	0.91
technology	technologie	0.82
technology	ethnology	0.81
technology	biotechnology	0.78
technology	webtechnology	0.78
technology	technologies	0.75
technology	terminology	0.73
technology	nanotechnology	0.71
technology	teknologi	0.69
technology	tech-blog	0.69

Figure 4.2: Syntactic similarity results

to it. This kind of situation can be avoided by applying more sophisticated similarity functions that exploit semantic knowledge to better discriminate.

Semantic similarity function. The semantic similarity function analyzes the semantic similarity of a pair of terms (t_x, t_y) considering their co-occurrence in the same term equipment as well as their frequency. The idea is that the more frequently two terms t_x and t_y co-occur, the more they are likely to be similar. In particular, the semantic similarity of two terms is directly proportional to the number of different term equipment both of them are included in. In fact, the more such number is high, the more the semantic relation between t_x and t_y can be considered to be valid in general, and not only dependent from the specific content of the web resource described by the web data item at hand. Moreover, the semantic similarity function also takes into account information coming from the frequency (i.e., the IDF value) of t_x and t_y within the WDI repository. The rationale is that we want to avoid to give too high importance to co-occurring terms which are very frequent. In particular, the semantic similarity of two terms t_x and t_y is inversely proportional to their frequency, and thus directly proportional to their respective IDF values. In fact, the more t_x and t_y rarely appear in the term collection, the more likely their co-occurrence denotes a semantic similarity between them.

In order to combine the information coming from term co-occurrence and fre-

quency, the semantic similarity function is defined as follows.

$$sim_{semantic}(t_x,t_y) = sim_{co-occurrence}(t_x,t_y) \cdot \left[\left(\frac{idf(t_x)}{MAX\ IDF} + \frac{idf(t_y)}{MAX\ IDF} \right) /2 \right]$$

Where the $sim_{co-occurrence}$ function evaluates the similarity deriving from the co-occurrence of t_x and t_y, $idf(t_x)$ and $idf(t_y)$ are the functions which evaluate the IDF value of t_x and t_y, respectively, and $MAXIDF$ is the IDF value of the term having the smallest frequency within the WDI repository.

The $sim_{co-occurrence}$ function is defined as follows.

$$sim_{co-occurrence}(t_x,t_y) = \frac{2 \cdot f(t_x,t_y)}{f(t_x)+f(t_y)}$$

Where $f(t_x,t_y)$ is the number of the term equipments including both t_x and t_y, $f(t_x)$ is the number of the term equipments including t_x, and $T(t_y)$ is the number of the term equipments including t_y. For each pair of terms (t_h,t_k), $sim_{co-occurrence}(t_x,t_y)$ normalizes the total number of co-occurrences of t_x and t_y against the total frequency of t_x and t_y independently.

The idf function is defined as follows.

$$idf(t_x) = -\log \frac{f(t_x)}{F}$$

Where $f(t_x)$ is the number of the term equipments including t_x and F is the frequency of the most recurring term in the WDI repository.

Example. The results of matching the term "technology" against the del.icio.us and Flickr datasets using the semantic similarity function are shown in Figure 4.3. With this kind of similarity, matching terms are more semantically related to "technology" than those returned by syntactic similarity, even if their similarity value with it is quite low. This is due to the fact that all matching terms are very frequent, and thus both the $sim_{co-occurrence}$ and the idf functions produce a rather low value.

Terminological similarity function. The terminological similarity function analyzes the terminological similarity of a pair of terms (t_x,t_y) by exploiting the WordNet relations (i.e., SYN, BT, NT, RT, IS) defined between them. The idea is to assess the

Keyword	Top-10 matching terms	Similarity value
technology	web	0.09
technology	computer	0.08
technology	geek	0.07
technology	internet	0.06
technology	software	0.05
technology	tech	0.05
technology	programming	0.04
technology	it	0.04
technology	news	0.04
technology	hardware	0.04

Figure 4.3: Semantic similarity results

similarity of two terms t_x and t_y on the basis of the kind of the terminological relations defined between t_x and t_y. To this end, a weight w is defined for each kind of terminological relation to assess its strength in determining the level of similarity, with $w_{SYN} \geq w_{BT} \geq w_{NT} \geq w_{IS} \geq w_{RT}$. Specific weights defined for terminological relations are:

- $w_{SYN} = 1.0$
- $w_{BT} = w_{NT} = w_{IS} = 0.8$
- $w_{RT} = 0.6$

Weights for terminological relationships have been borrowed from our HMatch 2.0 matching system [Castano et al., 2006] where they have been defined after extensive experimentation. We performed experimentations using them also on several term matching cases and we have seen that they work well also for term matching.

Formally, the terminological similarity function is defined as follows.

$$sim_{terminological}(t_x, t_y) = MAX\{w_{TR}\}$$

Where $TR \in \{SYN, BT, NT, RT, IS\}$ is a terminological relation.

Example. The results of matching the term "technology" against the del.icio.us and Flickr datasets using the terminological similarity function are shown in Figure 4.4. Using this kind of similarity in the similarity computation process provides as a re-

Keyword	Top-10 matching terms	Similarity value
technology	technologies	1.0
technology	engineering	1.0
technology	application	0.8
technology	applications	0.8
technology	nanotechnology	0.8
technology	computer+science	0.8
technology	computerscience	0.8
technology	biotechnology	0.8
technology	it	0.8
technology	hightech	0.8

Figure 4.4: Terminological similarity results

sult matching terms which are terminologically related with the target. In particular, the first result (i.e., "technologies") has the same lemma of "technology", and thus it belongs to the same equivalence cluster. The second result (i.e., "engineering") is a synonym of "technology". All remaining matching terms are either hypernyms (e.g., "application" , "applications") or hyponyms (e.g., "nanotechnology", "computer+science", "computerscience", "biotechnology", "it", "hightech") of "technology". The term "computer+science" is a compound word which is recognized in WordNet after the pre-processing step, replacing the special character "+" with a space. The other compound words (e.g., "nanotechnology", "computerscience", "biotechnology", "hightech") are recognized in WordNet after their tokenization in the respective components.

Linguistic similarity function. The linguistic similarity function determines the linguistic similarity of a pair of terms (t_x, t_y) by analyzing the linguistic relations between t_x and t_y. The idea is to consider t_x and t_y similar if t_x is an abbreviation or a substring of t_y or, vice versa, t_x is an extension or a compound form of t_y. Formally, the linguistic

similarity function is defined as follows.

$$sim_{linguistic}(t_x, t_y) = \begin{cases} 0.8 & if\ t_x\ is\ an\ abbreviation\ of\ t_y\ or\ t_y\ is\ an\ abbreviation\ of\ t_x \\ 0.6 & if\ t_x\ is\ a\ substring\ of\ t_y\ or\ t_y\ is\ a\ substring\ of\ t_x \end{cases}$$

The linguistic similarity function checks if t_x is an abbreviation of t_y, or vice versa, and if t_x is a substring of t_y, or vice versa, and returns a corresponding similarity value. If no linguistic relations exist between t_x and t_y, their linguistic similarity is set to zero. Otherwise, a constant value is returned depending on the kind of linguistic relation holding between t_x and t_y. In order to automatically find if a term is an abbreviation of another term, we rely on the on-line abbreviations dictionary Abbreviations.com[5]. We set the similarity value for the abbreviation relation higher than that of the substring relation to reflect a higher probability for t_x and t_y to be related terms in the former case. In fact, the substring relation can sometimes be misleading, as short terms can be included in many other terms, even if no real semantic connection exists between them.

Example. The results of matching the term "technology" against the del.icio.us and Flickr datasets using the linguistic similarity function are shown in Figure 4.5. The

Keyword	Top-10 matching terms	Similarity value
technology	tech	0.8
technology	tec	0.8
technology	it	0.8
technology	informationtechnology	0.6
technology	music_technology	0.6
technology	nanotechnology	0.6
technology	computer-technology	0.6
technology	computersandtechnology	0.6
technology	science_and_technology	0.6
technology	emerging-technology	0.6

Figure 4.5: Linguistic similarity results

application of this kind of similarity in the similarity computation process provides a set of matching terms which are compound or abbreviated forms of the keyword.

[5] http://www.abbreviations.com

4.2.2 Structural similarity

The structural similarity coefficient $ssim(wdi(wr_i), wdi(wr_j)) \in [0,1]$ of two web data items $wdi(wr_i)$ and $wdi(wr_i)$ is proportional to the number of fully matching properties in their corresponding structure equipments $\mathcal{S}(wr_i)$ and $\mathcal{S}(wr_j)$, as follows:

$$ssim(wdi(wr_i), wdi(wr_j)) = \frac{|\mathcal{FMP}|}{|\mathcal{MP}|}$$

where $\mathcal{MP}$ and $\mathcal{FMP}$ are the sets of *matching properties* and *fully matching properties* between the elements of the structure equipments $\mathcal{S}(wr_i)$ and $\mathcal{S}(wr_j)$, respectively. The set of matching properties $\mathcal{MP}$ is defined as

$$\mathcal{MP} = \left\{(p_h, p_k) \mid p_h[p^1] \sim p_k[p^1] \wedge p_h[p^n] \sim p_k[p^n] \wedge p_h[l] = p_k[l]\right\}$$

The set $\mathcal{MP}$ contains the pairs $\langle p_h, p_k \rangle$ of properties $p_h \in \mathcal{S}_i$ and $p_k \in \mathcal{S}_j$ that have similar property names (i.e., $p_h[p^1] \sim p_k[p^1] \wedge p_h[p^n] \sim p_k[p^n]$) and the same property path length (i.e., $p_h[l] = p_k[l]$). This choice is motivated by the fact that the same or very similar property names are frequently adopted by popular metadata formats, like the DC metadata initiative[6], the microformats[7], and the FOAF vocabulary[8]. The set of fully matching properties $\mathcal{FMP}$ is defined as

$$\mathcal{FMP} = \{(p_h, p_k) \mid \langle p_h, p_k \rangle \in \mathcal{MP} \wedge p_h[v] \sim p_k[v]\}$$

The set $\mathcal{FMP}$ contains the pairs of matching properties $\langle p_h, p_k \rangle \in \mathcal{MP}$ that have a similar property value (i.e., $p_h[v] \sim p_k[v]$). To detect whether $p_h[p^1] \sim p_k[p^1]$, $p_h[p^n] \sim p_k[p^n]$, and $p_h[v] \sim p_k[v]$, we rely on the similarity function *sim* described in Section 4.2.1 used in combination with a threshold-based mechanism.

4.2.3 Closeness coefficient evaluation

The closeness coefficient $CC(wdi(wr_i), wdi(wr_j))$ of two resources wr_i and wr_j is computed as the linear combination of the term and structural similarity coefficients of their corresponding wdi representations as follows:

$$CC(wdi_i, wdi_j) = w_{tsim} \cdot tsim(wdi_i, wdi_j) + (1 - w_{tsim}) \cdot ssim(wdi_i, wdi_j)$$

[6] http://dublincore.org

[7] http://microformats.org

[8] http://xmlns.com/foaf/spec

where the weight $w_{tsim} \in (0,1]$ denotes the impact of the term similarity in the evaluation of the overall closeness level. We note that also the logics equipment can be exploited to support the computation of the closeness coefficient when two Semantic Web resources wr_i and wr_j are involved. In this case, the types and class names contained in $\mathcal{L}(wr_i)$ and $\mathcal{L}(wr_j)$ can be used to execute a pre-matching phase. If $wdi(wr_i)$ and $wdi(wr_j)$ are instances of at least one common type/class, namely $\mathcal{L}(wr_i) \cap \mathcal{L}(wr_j) \neq \emptyset$, we consider these items as comparable and we continue with the computation of $CC(wdi(wr_i), wdi(wr_j))$ according to the above techniques, otherwise the matching execution is stopped and $CC(wdi_i, wdi_j)$ is set to 0. Moreover, we stress that the value of w_{tsim} can change from one matching execution to another with the aim to enforce a flexible configuration of the closeness computation and to tailor the calculation of the $CC(wdi(wr_i), wdi(wr_j))$ coefficient according to the specific kind of web data items to match. A default value of w_{tsim} is automatically set in the closeness computation according to the different, possible matching cases that can occur (see Table 4.2).

	Tagged resource	**Microdata resource**	**Semantic Web resource**
Tagged resource	$w_{tsim} = 1.0$	$w_{tsim} = 1.0$	$w_{tsim} = 1.0$
Microdata resource	$w_{tsim} = 1.0$	$0.6 \leq w_{tsim} \leq 0.9$	$0.4 \leq w_{tsim} \leq 0.6$
Semantic Web resource	$w_{tsim} = 1.0$	$0.4 \leq w_{tsim} \leq 0.6$	$0 < w_{tsim} \leq 0.5$

Table 4.2: The default weight w_{tsim} for different kinds of web data items to match

When a tagged resource is involved in the matching case, the default w_{tsim} value is set to 1.0. This is due to the fact that the wdi representation of tagged resources are only characterized by term equipments and the closeness evaluation can be exclusively based on the linguistic similarity. When two microdata resources are matched, we observe that the corresponding wdi representation are characterized by rich term equipments built upon the textual content of the resource. At the same time, we note that the structure equipment is usually not very meaningful due to the fact that the properties of microdata are limited in number and rarely concern the resource topic. For this reason, the w_{tsim} weight is kept quite high in the range $0.6 \leq w_{tsim} \leq 0.9$ to assign a high impact to linguistic similarity with respect to structural similarity in the overall closeness evaluation. A value $0.4 \leq w_{tsim} \leq 0.6$ is adopted when a microdata resource is matched against a Semantic Web resource. This choice allows to obtain satisfactory closeness values when $wdi(wr_i)$ and $wdi(wr_j)$ are similar from the linguistic point of

view, apart from their structural similarity *ssim*. Moreover, this weight ensures a higher closeness value when $wdi(wr_i)$ and $wdi(wr_j)$ present a similar structure in addition to a high linguistic similarity. Term and structure equipments can be considered as equally meaningful when two Semantic Web resources are matched. In some cases, due to the fact that the specification of a Semantic Web resource is usually composed of a number of different properties, the structure equipment can be considered more relevant for matching than the term equipment itself. For this reason, a weight $0 < w_{tsim} \leq 0.5$ is suggested for these matching cases.

Running example. We consider the Semantic Web resource $freebase1$ shown in Figure 3.3(c) describing the movie Star Wars Episode IV - A New Hope and the following $freebase4$ describing Star Wars Episode I - The Phantom Menace:

$\mathcal{T} =$ $\{(1999,1),(episode,1),(directed,1),(fiction,1),(film,1),(ford,1),(george,1),$
$(harrison,1),(i,1),(lucas,1),(menace,1),(name,3),(phantom,1),(science,1),$
$(star,1),(starring,1),(the,1),(year,1),(wars,1)\}$

$\mathcal{S} =$

p1	**pn**	**v**	**l**
name	*name*	Star Wars...	1
year	*year*	1999	1
directed_by	*name*	George Lucas	2
starring_in	*name*	Harrison Ford	2

$\mathcal{L} =$ $\{Film, Science\ Fiction\}$

$prov =$ `http://www.freebase.com/view/en/star_wars_episode_i_the_phantom_menace`

We choose to show a matching example of two Semantic Web resources since this kind of resource is suitable to present both the term and the structural matching techniques we employ in the computation of the closeness coefficient. The term equipments $\mathcal{T}(freebase1)$ and $\mathcal{T}(freebase4)$ differs in the terms appearing in the title of the two movies and in the year of their commercial distribution. The term similarity coefficient $tsim(wdi(freebase1), wdi(freebase4))$ is calculated as follows:

$$tsim(wdi(freebase1), wdi(freebase4)) = \frac{2 \cdot |t_x \sim t_y|}{|\mathcal{T}(freebase1)| + |\mathcal{T}(freebase4)|} = 0.7$$

The structure equipments $\mathcal{S}(freebase1)$ and $\mathcal{S}(freebase4)$ differs in the value of the first two properties, thus $wdi(freebase1)$ and $wdi(freebase4)$ have four matching

properties (i.e., *name*, *year*, *directed_by*, *starring_in*) and two fully matching properties (i.e., *directed_by*, *starring_in*). The structural similarity coefficient is then the following:

$$ssim(wdi(freebase1), wdi(freebase4)) = 0.5$$

By setting $w_{tsim} = 0.5$, the closeness coefficient $CC(wdi(freebase1), wdi(freebase4))$ is equal to 0.6.

Chapter 5

Construction of *in*-clouds

An *in*-cloud is a collection of web data items relevant to a specific target entity. Its aim is to organize the answers to a query in a structure showing the *prominence* of each retrieved web resource with respect to the target entity and the level of *closeness* between each pair of retrieved web resources. The *in*-cloud construction is performed in two steps: the classification of the web resources, described in Section 5.1 and the clouding of web resources, described in Section 5.2.

5.1 Classification of web resources

Our semantic clouding approach is based on the capability of grouping the retrieved web resources on the basis of their closeness. The closeness between two web data items i and j captures the level of similarity/semantic relation holding between them and it is represented by a closeness coefficient $CC(wdi(wr_i), wdi(wr_j)) \in [0,1]$, calculated by considering their respective wdi representations $wdi(wr_i)$ and $wdi(wr_j)$. Through the matching techniques described in Section 4.2, the closeness coefficient $CC(wdi(wr_i), wdi(wr_j))$ is calculated for any possible pair of web data items stored in the WDI repository. These coefficients are kept in a *closeness matrix M* of dimension k, where k is the number of web data items in the WDI repository.

5.1.1 Clustering procedure

Starting from the closeness matrix M, a *hierarchical* clustering technique of *agglomerative* type is employed [Castano et al., 2001]. Agglomerative refers to the property

of the technique to proceed by a series of successive merging of web data items into groups. Hierarchical refers to the property of the technique to classify linked data items into groups at different levels of closeness to form a tree. This choice is motivated by the fact that the agglomerative hierarchical clustering follows a bottom-up approach and thus the cluster computation can be stopped once clusters of the desired level of closeness are defined. This is suitable for semantic clouding, where we are interested in finding candidate clusters of prominent web data items where a minimum level of closeness is required (see Section 5.2).

The hierarchical clustering procedure $HC(M)$ is shown in Figure 5.1.

Procedure *HC*
Input: closeness matrix M
Output: a closeness tree CT

Let k be the number of web data items in M
for $i := 1$ to k **do**
  $M[i,i] := 1$
  for $j := 1$ to k **do**
    $M[j,i] := M[i,j] := CC(wdi(wr_i), wdi(wr_j))$
  end for
end for
for all $wdi(wr_i) \in M$ **do**
  Put $wdi(wr_i)$ into a cluster Cl_i
end for
while $k > 1$ **do**
  Select $Cl_i, Cl_j \mid M[i,j] = max_{s,t}(M[s,t])$
  $Cl_i = Cl_i \cup Cl_j$
  for $l := 1$ to k **do**
    if $l \neq j$ **then**
      $M[i,l] := M[l,i] := min(M[l,i], M[l,j])$
    end if
    Update M by deleting row and column corresponding to Cl_j
  end for
  $k := k - 1$
end while

Figure 5.1: The hierarchical clustering procedure *HC*

Given a closeness matrix M of dimension k, the hierarchical clustering procedure $HC(M)$ first creates a cluster Cl_i for each web data item $wdi(wr_i)$ that appears in M. Then, the two clusters Cl_i and Cl_j having the highest closeness coefficient in M are selected and their corresponding clusters Cl_i and Cl_j are merged. The merge operation is performed by taking the union of the two selected clusters Cl_i and Cl_j, that is $Cl_i = Cl_i \cup Cl_j$. The row and the column of the newly defined cluster Cl_i is updated in M by determining the closeness coefficient values between Cl_i and each remaining cluster Cl_l in M. To calculate the closeness coefficient value between two clusters Cl_i and Cl_l, with $Cl_l \neq Cl_j$, it is possible to rely on three different strategies, namely the *complete-link* strategy, *single-link* strategy, and the *average-link* strategy. Using the complete-link strategy [Manning et al., 2008], the closeness coefficient value between Cl_i and Cl_l (i.e., $M[i,l] = M[l,i]$ in the closeness matrix M) is calculated as the minimum closeness coefficient holding between Cl_i and Cl_l (i.e., $M[i,l] = M[l,i]$) and between Cl_j and Cl_l (i.e., $M[j,l] = M[l,j]$), that is $M[i,l] = M[l,i] = min\{M[l,i], M[l,j]\}$. With the single-link strategy, the closeness coefficient value between Cl_i and Cl_l is calculated as the maximum closeness coefficient holding between Cl_i and Cl_l and between Cl_j and Cl_l, that is $M[i,l] = M[l,i] = max\{M[l,i], M[l,j]\}$. Finally, the average-link strategy calculates the closeness coefficient value between Cl_i and Cl_l as the average closeness coefficient holding between Cl_i and Cl_l and between Cl_j and Cl_l, that is $M[i,l] = M[l,i] = \frac{M[l,i]+M[l,j]}{2}$. The complete-link strategy produces a higher number of smaller clusters than the single and the average-link strategies, but these clusters are more cohesive since a minimum level of closeness is ensured to any pair of web data items in a given cluster Cl_i. After the computation of the closeness coefficient holding between the new cluster and all the others, the row and the column of the cluster Cl_j is deleted from the matrix M. The clustering procedure terminates when the dimension of M is 1. The result of the clustering procedure is a *closeness tree* CT where each leave corresponds to a web data item, while intermediate nodes represent virtual elements, (i.e., cluster "centroids" [Salton, 1989]) which are represented by a closeness coefficient value in the tree. Several clusters can be identified in CT, depending on the value of the closeness coefficient. In the following, a cluster Cl_i with an associated closeness coefficient value CC in the closeness tree is denoted Cl_i^{CC}.

Running example. In Figure 5.2, we show an example of closeness tree resulting from the execution of the hierarchical clustering procedure $HC(M)$ (relying on the

complete-link strategy) on the web data items of Figure 3.1.

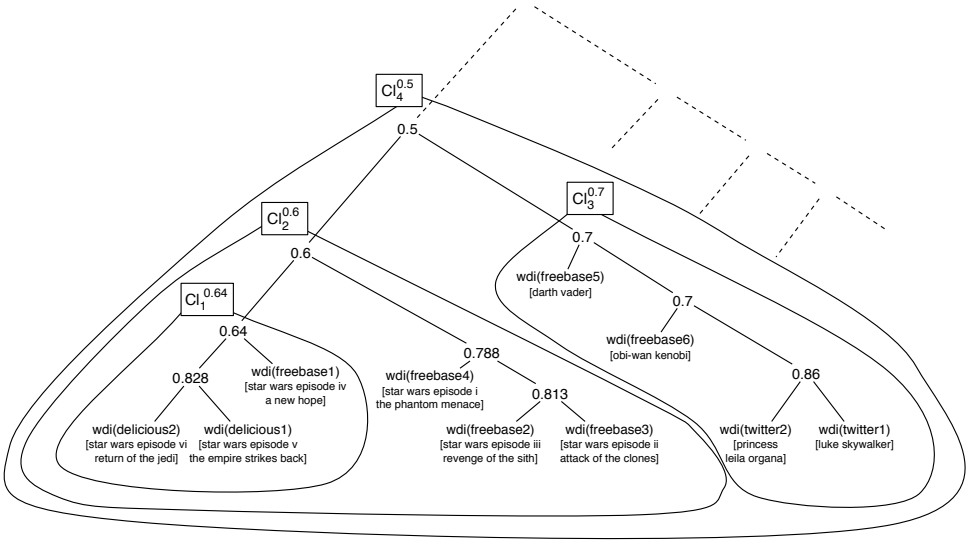


Figure 5.2: Example of a portion of closeness tree and clusters of web data items

In Figure 5.2, an example of possible clusters is also highlighted. For instance, the cluster $Cl_2^{0.6}$ is characterized by a closeness coefficient value of 0.6.

5.2 Clouding of web resources

An *in*-cloud is built out of the closeness tree CT by properly arranging the web data items prominent for a considered target entity e. Formally, an *in*-cloud $\mathcal{IC}_e$ is defined as follows.

Definition: *in*-cloud. *An in-cloud is an undirected weighted graph $\mathcal{IC}_e = (N,E)$ associated with a target entity e, where N is the set of nodes of $\mathcal{IC}_e$, and E is the set of edges of $\mathcal{IC}_e$. In particular, a node $n_i \in N$ represents a web data item $wdi(wr_i)$. An edge $(e_h, e_k) \in E$ between two nodes n_h and n_k is labeled with the level of closeness holding between the corresponding web data items $wdi(wr_h)$ and $wdi(wr_k)$. $\mathcal{IC}_e$ is equipped with a labeling function $\rho : N \rightarrow (0,1]$, that associates each node $n_i \in N$ with a value $\rho(n_i)$ in the range $(0,1]$, and a labeling function $\sigma : E \rightarrow (0,1]$, that associates*

each edge $(e_h, e_k) \in E$ *with a value* $\sigma(e_h, e_k)$ *in the range* $(0, 1]$. *A value* $\rho(n_i)$ *denotes the level of prominence of the corresponding web data item* $wdi(wr_i)$ *in* $\mathcal{IC}_e$. *A high value of* $\rho(n_i)$ *denotes a high prominence of the web resource represented by the web data item* $wdi(wr_i)$ *in the* in-*cloud. A value* $\sigma(e_h, e_k)$ *denotes the level of closeness between the web data items* $wdi(wr_h)$ *and* $wdi(wr_k)$ *in* $\mathcal{IC}_e$. *In particular,* $\sigma(e_h, e_k)$ *is equal to the closeness value* $CC(wdi(wr_h), wdi(wr_k))$ *of the cluster* Cl^{CC} *containing both* $wdi(wr_h)$ *and* $wdi(wr_k)$ *in the closeness tree CT*.

Given the target entity e, the construction of the *in*-cloud $\mathcal{IC}_e$ is articulated in three steps as shown in Figure 5.3.

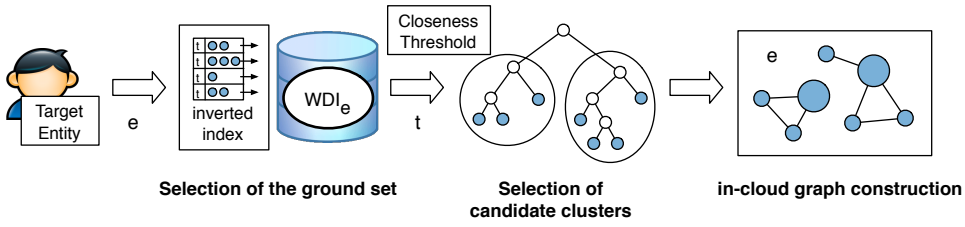


Figure 5.3: *in*-cloud construction workflow

1. Selection of the ground set. In this step, the web data items composing the ground set WDI_e of e are extracted from the WDI repository. To this end, the specification of the target entity e is tokenized if necessary into a set K of keywords and each keyword $k_i \in K$ is searched in the inverted index of the WDI repository. If an index term it_i is retrieved such that $it_i = k_i$, all the web data items associated with it_i in the inverted index are added to WDI_e.

2. Selection of candidate clusters. Candidate clusters to be used for the *in*-cloud graph construction are selected from the closeness tree CT (see Section 5.1). Candidate clusters to build an *in*-cloud $\mathcal{IC}_e$ are selected within CT using a threshold-based mechanism, starting from the web data items in WDI_e. A candidate cluster is defined as follows.

Definiton: Candidate Cluster. *Given the closeness tree CT, a* $wdi(wr_i) \in WDI_e$ *and the closeness threshold* $t \in (0,1]$*, a cluster* Cl_j^{CC} *in CT is a candidate cluster, denoted by* $\overline{Cl_j^{CC}}$*, if and only if:*

- $wdi(wr_i) \in Cl_j^{CC}$
- *The centroid of* Cl_j^{CC} *is the highest ancestor of* $wdi(wr_i)$ *such that* $CC \geq t$

Candidate clusters can be described in terms of their *size* and their *level of homogeneity*. The size of a candidate cluster $\overline{Cl_j^{CC}}$ is equal to the number of web data items contained in $\overline{Cl_j^{CC}}$. The level of homogeneity is given by the closeness coefficient labeling the root node of $\overline{Cl_j^{CC}}$ (i.e., CC). Size and level of homogeneity depend on the choice of the threshold t. High values of t produce small clusters with high homogeneity, while low values of t produce large clusters with a lower level of homogeneity. The number of candidate clusters that are selected depends instead on both the number of web data items in WDI_e and the threshold t, and the number of web data items in WDI_e depends in turn on the target entity e. A generic target entity e, together with a low value of the closeness threshold t, will produce few but very large and non-homogeneous clusters. The same situation, but with a high value of t, will produce a (very) high number of small and homogeneous clusters. By selecting a more specific target entity, it is possible to reduce the number of web data items in WDI_e and, consequently, the number of candidate clusters.

3. *in*-cloud graph construction. The *in*-cloud graph $\mathcal{IC}_e = (N,E)$ for the target entity e is created by iterating the procedure of Figure 5.4 for each candidate cluster $\overline{Cl_j^{CC}}$ selected in the previous step.

The procedure creates a node $n_i \in \mathcal{IC}_e$ for each web data item in a candidate cluster $\overline{Cl_i^{CC}}$. Then, the closeness tree $\overline{Cl_i^{CC}}$ is visited starting from the leaves, and each internal node *in* of $\overline{Cl_i^{CC}}$ is analyzed, in order to discover the edges that should be inserted in the *in*-cloud $\mathcal{IC}_e$. To this end, for each visited *in*, a set of (set of) leaves is selected. The cardinality of such set is equal to the degree of $\overline{Cl_i^{CC}}$; thus, in the case of binary trees, it is equal to 2. Formally, if d is the degree of $\overline{Cl_i^{CC}}$, a set LCS (Leaf Children Set) is created for each *in* as follows:

$$LCS = \left\{ S^1, \ldots, S^d \right\}$$

Procedure *GC*
Input: a candidate cluster $\overline{Cl_i^{CC}}$
Output: an *in*-cloud $\mathcal{IC}_e$ for the target entity e

Insert a node in N for all the leaves in $\overline{Cl_i^{CC}}$
for all level of depth i of $\overline{Cl_i^{CC}}$ (from $i = h$ to $i = 0$, with h = height of $\overline{Cl_i^{CC}}$) **do**
  for all internal node *in* of $\overline{Cl_i^{CC}}$ **do**
    for all descending edge j of *in* (from $j = 1$ to $j = d$, with d = degree of $\overline{Cl_i^{CC}}$) **do**
      Select the child (children) of *in* which is a leaf (are leaves) and is (are) at the lowest level of $\overline{Cl_i^{CC}}$
    end for
  end for
  Insert an edge in E for each pair in the Cartesian product of the selected children, labeling each one with the value of *in*
end for

Figure 5.4: The procedure *GC* for *in*-cloud graph construction from a candidate cluster $\overline{Cl_i^{CC}}$

where each S^i, with $1 \leq i \leq d$, is defined as the set of children of *in* descending from its i-th edge that are leaves and that are at the same lowest level of $\overline{Cl_i^{CC}}$. Finally, an edge is inserted in $\mathcal{IC}_e$ for each pair of nodes (n_i, n_j) in the Cartesian product of the selected children, where $n_i \in S^h$ and $n_j \in S^k$, with $h \neq k$, and each one of them is labeled with the value of *in*.

The graph $\mathcal{IC}_e$ is finally labeled through the labeling function ρ to assign levels of prominence to nodes. Different prominence evaluation techniques that can be employed are described in Section 5.2.1.

We note that the presented procedure can be applied to generic trees, not only binary trees. It can be useful to execute the clouding over generic input tree, not only generated by the clustering process described in Section 5.1.

Running example. As an example of *in*-cloud creation, we take into account the closeness tree shown in Figure 5.2. Then, we start from the target entity "Star Wars"

and we choose a closeness threshold of 0.5. The resulting ground set $WDI_{StarWars}$ is composed by the web data items denoting the movies of the Star Wars saga $wdi(delicious1)$, $wdi(delicious2)$, $wdi(freebase1)$, $wdi(freebase2)$, $wdi(freebase3)$, and $wdi(freebase4)$, because the target entity is directly retrieved in their term equipment. Then, we use the closeness threshold 0.5 to select the candidate clusters. In our example, we select only one cluster $Cl_4^{0.5}$ that contains all the web data items in $WDI_{StarWars}$ plus the web data items $wdi(twitter1)$, $wdi(twitter2)$, $wdi(freebase5)$, and $wdi(freebase6)$ that represent web data items about the main characters of the movie saga. Finally, the candidate cluster $Cl_4^{0.5}$ is submitted to the *in*-cloud graph creation procedure (see Figure 5.4). The procedure creates a node in the *in*-cloud graph $\mathcal{IC}_{StarWars}$ for each web data item in $Cl_4^{0.5}$. Then, the internal nodes at the highest level of depth of $Cl_4^{0.5}$ are analyzed. In this step, we create the edges $(wdi_{delicious1}, wdi_{delicious2})$, $(wdi_{freebase2}, wdi_{freebase3})$ and $(wdi_{twitter1}, wdi_{twitter2})$, which are labeled with the value of their parent node, respectively (i.e., 0.828, 0.813, 0.86). Then, the internal nodes at a lower level of depth are analyzed, creating an edge between $wdi(freebase1)$ and $wdi(delicious1)$, and between $wdi(freebase1)$ and $wdi(delicious2)$, both labeled with the closeness value 0.64. Analogously, $wdi(freebase4)$ is connected with $wdi(freebase2)$ and $wdi(freebase3)$, and $wdi(freebase6)$ is connected with $wdi(twitter1)$ and $wdi(twitter2)$. At a further lower level of depth, an edge labeled with the closeness value 0.6 is created between $wdi(freebase1)$ and $wdi(freebase4)$, and an edge labeled with the closeness value 0.7 is created between $wdi(freebase5)$ and $wdi(freebase6)$. Finally, $wdi(freebase5)$ is connected with $wdi(freebase1)$ and $wdi(freebase4)$ through an edge labeled with the closeness value 0.5. The resulting *in*-cloud graph is shown in Figure 5.5 together with the edge labels representing closeness. Prominence is instead discussed in the following.

5.2.1 Prominence evaluation

Different techniques are possible for the evaluation of the web data items prominence in an *in*-cloud and these techniques can be used alone or in combination. We devise three main categories of techniques for prominence evaluation, namely *provenance-based*, *target-based*, and *popularity-based* techniques.

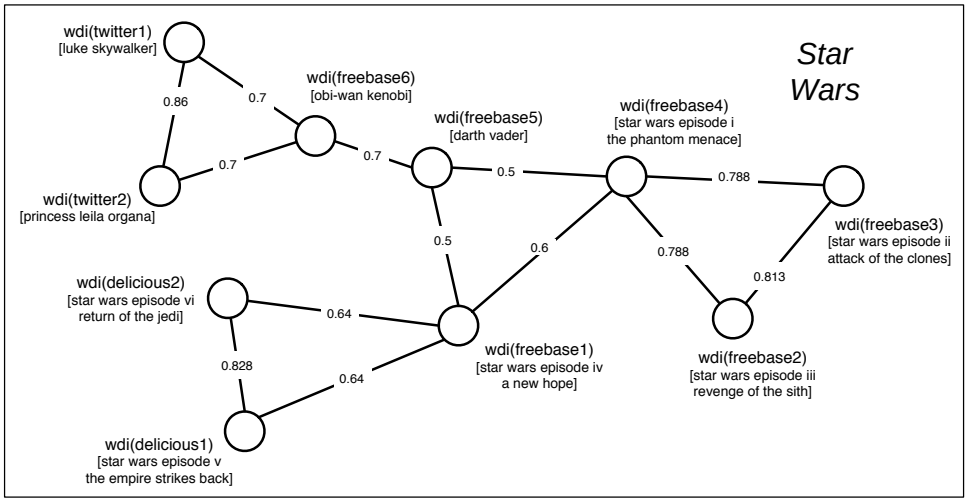


Figure 5.5: Example of a *in*-cloud graph derived from candidate cluster $Cl_4^{0.5}$ of Figure 5.2

Provenance-based techniques. Provenance-based techniques for prominence evaluation are based on the idea that the prominence $\rho(wdi(wr_i))$ in an *in*-cloud corresponds to the level of trust and importance of the data source from which $wdi(wr_i)$ has been extracted. This idea relies on the fact that some sources on the web can be considered more "authoritative" than others with respect to a certain information request. The prominence value of a data source can be automatically set by exploiting techniques based on measurable features of the data source, such as link analysis, language analysis, and reliability of the source in time [Kleinberg, 1999; Gil and Artz, 2007]. Alternatively, the prominence value can be manually refined by an user to personalize the relevance of each data source according to her/his expertise in the domain. Moreover, provenance-based techniques may be target-dependent or target-independent. For target-dependent techniques, the target entity e associated with $\mathcal{IC}_e$ is taken into account for determining the prominence value of the web data items. This means that the prominence of a data source can vary from one *in*-cloud to another according to the associated target entity. On the opposite, target-independent techniques are based on the idea that data sources are authoritative per sé, independently from the considered target entity. In this case, the prominence value of a data source is computed before the *in*-cloud creation and it remains fixed.

We calculate the provenance-based prominence value $\rho_{provenance}(wdi(wr_i))$ of a web data item $wdi(wr_i)$ as follows:

$$\rho_{provenance}(wdi(wr_i)) = \frac{w_{source} \cdot \#\ of\ inbound\ links\ of\ wdi(wr_i)\ in\ source}{MAX\ PROMINENCE}$$

where w_{source} is the weight associated with a web source *source*. In particular, we associate a weight of 1 with Freebase, a weight of 0.7 with Twitter, and a weight of 0.5 with del.icio.us. Such weight is then multiplied for the number of inbound links that the web data item $wdi(wr_i)$ has within the web source *source*. Finally, the provenance value $\rho_{provenance}(wdi(wr_i))$ is normalized with the highest provenance-based prominence value in the *in*-cloud including the web data item $wdi(wr_i)$.

Target-based techniques. Target-based techniques for prominence evaluation are based on the idea that the prominence $\rho(wdi(wr_i))$ in an *in*-cloud depends on the fact that $wdi(wr_i)$ has a direct or an indirect relation with the target entity e. A $wdi(wr_i)$ has a direct relation with e if it contains one or more terms of e, that is, if $wdi(wr_i)$ directly appears in the ground set WDI_e (i.e., $wdi(wr_i) \in WDI_e$). In such a case, the prominence value $\rho(wdi(wr_i))$ is set to 1. A $wdi(wr_i)$ has an indirect relation with e if there is a closeness path $P = wdi(wr_i) \leftrightarrow wdi(wr_{i+1}) \leftrightarrow \cdots \leftrightarrow wdi(wr_{j-1}) \leftrightarrow wdi(wr_j)$ in $\mathcal{IC}_e$ with a $wdi(wr_j) \in WDI_e$. In such a case, the prominence value $\rho(wdi(wr_i))$ is computed as the product of all the closeness coefficients $CC(wdi(wr_k, wdi(wr_{k+1})$ labeling the edges in the closeness path P. If $wdi(wr_i)$ has an indirect relation with e through more than one closeness path, the shortest path P_{min}.

Thus, we calculate the target-based prominence value $\rho_{target}(wdi(wr_i))$ of a web data item $wdi(wr_i)$ as follows:

$$\rho_{target}(wdi(wr_i)) = \frac{\prod_{k=i}^{j-1} CC(wdi(wr_k), wdi(wr_{k+1}))}{MAX\ PROMINENCE} \,|(wdi(wr_k), wdi(wr_{k+1})) \in P_{min}$$

where $\prod_{k=i}^{j-1} CC(wdi(wr_k), wdi(wr_{k+1}))$ is the weighted product of the closeness coefficients associated with the edges in the shortest path P_{min} between $wdi(wr_i)$ and $wdi(wr_j)$, where $wdi(wr_j) \in WDI_e$. Finally, the provenance value $\rho_{target}(wdi(wr_i))$ is normalized with the highest target-based prominence value in the *in*-cloud including the web data item $wdi(wr_i)$.

Popularity-based techniques. Popularity-based techniques for prominence evaluation are based on the idea that the prominence $\rho(wdi(wr_i))$ in $\mathcal{IC}_e$ is calculated by analyzing the topology of the *in*-cloud graph, on the basis of the degree of connection of the nodes therein contained. Techniques based on the graph topology for evaluating the popularity of users in social networks have been recently received a lot of attention in the literature [Easley and Kleinberg, 2010], and they can be suitably adopted also for popularity-based prominence evaluation. In this context, a common measure is based on the notion of *degree of centrality*. According to this measure, the prominence (i.e., popularity) of a node $\rho(wdi(wr_i))$ is proportional to the number of edges that involve a given node $wdi(wr_i)$. The closeness coefficients of the edges in $\mathcal{IC}_e$ can be also considered to weight the strength of each edge in the overall computation of the degree of centrality of a node. Other common measures of centrality are the *betweenness centrality* [Newman, 2005] and the *closeness centrality* [Goh et al., 2003], which take into account the global structure of the *in*-cloud graph for calculating the prominence of a node. By relying on the betweenness centrality measure, the prominence $\rho(wdi(wr_i))$ of a node $wdi(wr_i)$ is proportional to the number of shortest paths between $wdi(wr_i)$ and all the other nodes of $\mathcal{IC}_e$. The higher is the number of shortest paths between $wdi(wr_i)$ and the other nodes, the higher is the prominence/centrality of $wdi(wr_i)$. On the opposite, the prominence $\rho(wdi(wr_i))$ based on the closeness centrality measures the distance of $wdi(wr_i)$ from the other nodes of $\mathcal{IC}_e$ and it can be computed according to a number of different measures of distance [Opsahl et al., 2010].

We calculate the popularity-based prominence value $\rho_{popularity}(wdi(wr_i))$ of a web data item $wdi(wr_i)$ as follows:

$$\rho_{popularity}(wdi(wr_i)) = \frac{\sum CC(wdi(wr_i), wdi(wr_j))}{MAX\ PROMINENCE}$$

where $\sum CC(wdi(wr_i), wdi(wr_j))$ is the weighted sum of the closeness coefficients associated with the edges between $wdi(wr_i)$ and each other web data item $wdi(wr_j)$ in the *in*-cloud. Finally, the provenance value $\rho_{popularity}(wdi(wr_i))$ is normalized with the highest popularity-based prominence value in the *in*-cloud including the web data item $wdi(wr_i)$.

Running example. We refer to the *in*-cloud graph of Figure 5.5 that is transformed into the *in*-cloud shown in Figure 3.1 by exploiting a popularity-based technique for

prominence evaluation based on degree centrality. An example of an *in*-cloud where prominence is evaluated according to the target-based techniques is shown in Figure 5.6.

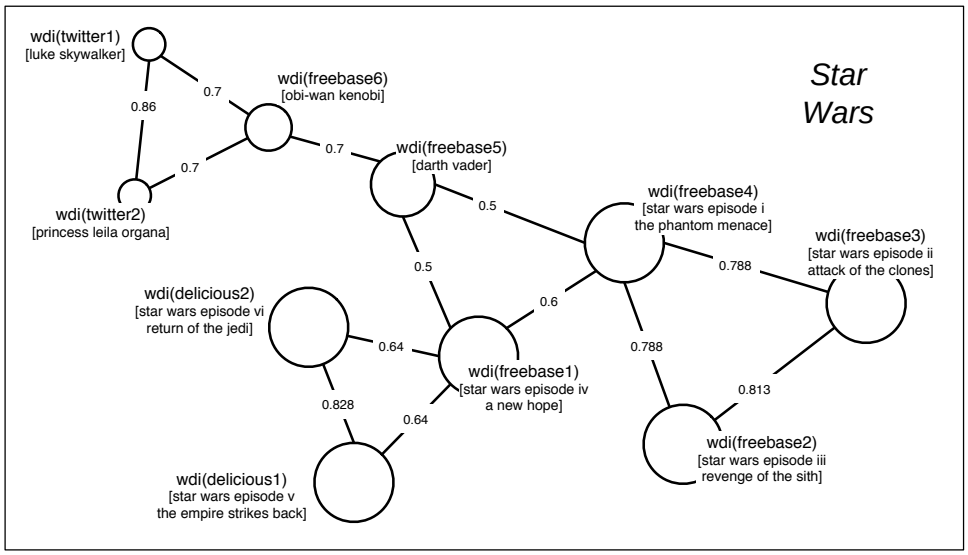


Figure 5.6: Example of *in*-cloud where prominence has been evaluated according to target-based techniques

In this case, the most prominent nodes are those representing web data items directly matching the target entity "Star Wars". For the other nodes, the level of prominence decreases as much as they are far from the nodes directly matching the target entity.

5.3 Comparison between *in*-clouds, Linked Data, and Wolfram Alpha

In this section, we show a comparison between *in*-clouds, Linked Data, and Wolfram Alpha, and we analyze the differences in the output obtained by specifying the same target entity, namely "Star Wars". In particular, we compare the output obtained using our clouding techniques with a Linked Data portion related to the target entity and with

the output obtained using the computational knowledge engine Wolfram Alpha. A detailed evaluation about the users' perception of the *in*-clouds effectiveness is provided in Section 6.1.

5.3.1 Linked Data vs *in*-clouds

As described in Section 2.2, Linked Data aims at linking together different descriptions of the same real-world object, stored in different RDF repositories available on the web. As an example, in Figure 5.7, we show a small portion of Linked Data related to the target entity "Star Wars".

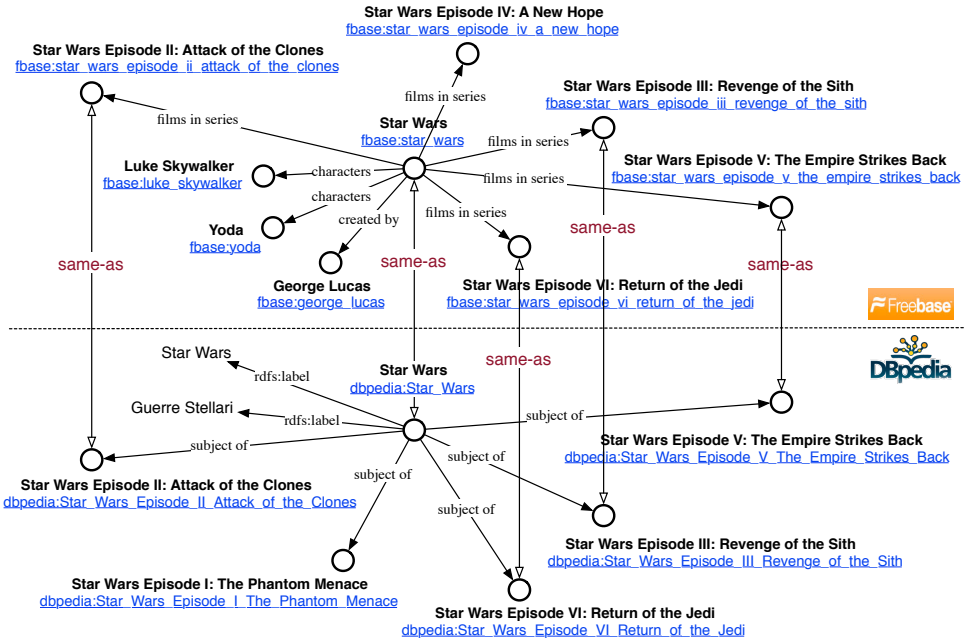


Figure 5.7: A portion of Linked Data related to the target entity "Star Wars"

The portion of Linked Data represented in Figure 5.7 shows the Semantic Web resources which are connected to the target entity "Star Wars" in the repositories Freebase and DBpedia. Similarly to *in*-clouds, it provides information about all the related Star Wars saga movies, as well as all the main characters. However, Linked Data and *in*-clouds differ in many aspects, which are summarized in Table 5.1.

Linked Data	***in*-cloud**
Resulting structure: graph	Resulting structure: graph
Aim: connect different RDF descriptions of the same object	Aim: organize the relevant web resources for a target entity
Off-line process	On-line process
One general graph (connecting different repositories)	One graph for each target entity
Directed graph	Undirected graph
Unweighted graph	Weighted graph
The nodes can be URIs or literals	The nodes are web data items
The edges can be labeled with properties or with *owl:sameAs*	The edges are labeled with the value of closeness between web data items
Connected data are described using RDF	Connected data are described using the WDI model
No distinction between the nodes	Each node has a different prominence
Only descriptions referred to the same object are connected	Similar web data items are connected by different closeness values
Data which are not described using RDF cannot be included	Each kind of web resource can be included

Table 5.1: Comparison between Linked Data and *in*-clouds

The main difference between Linked Data and *in*-clouds is that Linked Data does not take into account the Social Web resources, such as the users' comments, posts and personal feeds, while *in*-clouds include both Social and Semantic Web resources. Moreover, Linked Data builds a flat graph structure of interconnected URIs, while in an *in*-cloud the retrieved web resources are organized on the basis of their prominence with respect to the target entity and of their reciprocal closeness.

5.3.2 Wolfram Alpha vs *in*-clouds

Wolfram Alpha[1] is a computational knowledge engine which is able to answer questions, do math and calculus, convert money, make statistics, and create plots and visualizations. The available information includes vast scientific, technical, chemical, medical, health, business, financial, weather, and geographic data, but it provides a limited support for social knowledge.

In Figure 5.8, the output produced by Wolfram Alpha in response to the query "Star Wars" is shown.

As we can see, the output produced by Wolfram Alpha is very different from the *in*-cloud produced for the target entity "Star Wars". In fact, Wolfram Alpha provides a simple list of related Star Wars saga movies, and provides no information about the main characters. In addition, it shows some statistics about the box office receipts and the number of screens related to the different Star Wars episodes. The main differences

[1] `http://www.wolframalpha.com`

Figure 5.8: Output produced by Wolfram Alpha in response to the query “Star Wars”

between Wolfram Alpha and *in*-clouds are summarized in Table 5.2.

The main difference between Wolfram Alpha and *in*-clouds is that, as Linked Data, Wolfram Alpha does not take into account the Social Web resources, while *in*-clouds include both Social and Semantic Web resources. In particular, Wolfram Alpha is able to answer queries only providing focused and objective information, by extracting information only from structured knowledge repositories. On the other hand, in our se-

Wolfram Alpha	***in*-cloud**
Resulting structure: statistics, plots, formula, ...	Resulting structure: graph
Aim: answer a specific question	Aim: provide a comprehensive set of relevant information for a target entity
Context: specific to the question (pointed answer)	Context: extended to related facts (exploratory answer)
On-line process	On-line process
Focus on scientific knowledge	Focus on general knowledge
Only objective information is provided	Both objective and subjective information is provided
Information are extracted only from structured knowledge repositories	Each kind of web source is considered
Provenance of retrieved information is not provided to the user	Provenance of retrieved information is provided to the user

Table 5.2: Comparison between Wolfram Alpha and *in*-clouds

mantic data clouding approach, each kind of Web is analyzed, considering the Social Web, the Semantic Web, and the Web of Documents, and thus providing both objective and subjective information about the target entity. A further difference between Wolfram Alpha and *in*-clouds concerns the fact that the provenance of information retrieved by Wolfram Alpha is not shown to the user, while in an *in*-cloud the provenance of each web resource is provided, so that the user can explore directly the retrieved web resources.

Chapter 6

Evaluation issues

In this chapter, we show and discuss the results of all the evaluation experiments we have conducted to test the different techniques introduced in this thesis, as well as the whole system.

In Section 6.1, we present the user evaluation about the perceived effectiveness of *in*-clouds. Such evaluation has been conducted over a group of Computer Science students, and aims at comparing the *in*-clouds with the other available tools on the web. In Section 6.2, we evaluate our semantic clouding approach as a whole. In particular, we evaluate the system accuracy, the cohesion of the produced *in*-clouds, and the system scalability.

6.1 User evaluation

Hypothesis. In this section, we evaluate the user perception and the degree of user satisfaction about *in*-clouds as a tool for exploration and browsing of web resources.

Experimental setup. Our approach to user evaluation of *in*-clouds is based on user-oriented evaluation methods that have been proposed in the literature for interactive web search interfaces and systems [Hoeber and Yang, 2007; Leclercq, 2007]. These methods recommend to focus on a group of users (usually small) who are asked to perform specific web search tasks using one or more software systems. Then, the same users are asked to fill in a questionnaire concerning their search experience. For

in-cloud evaluation, we designed a questionnaire and we identified a group of 18 students of the Databases course of the Master Degree in Computer Science held at the University of Milan. Such students had a similar background on Linked Data and Semantic Web, mainly based on a few classes delivered on these topics in the course. This group of students was required to work on three test cases corresponding to different kinds of target entities, web resources, and data sources, as summarized in Table 6.1.

Table 6.1: Test cases for user evaluation

	Target entity	**Web resources**	**Data sources**
Test case 1	Mac OS X	Microdata Semantic Web	Twitter Freebase
Test case 2	Star Wars	Semantic Web	Freebase, DBpedia DBpedia
Test case 3	London Olympics	Microdata Semantic Web	RSS channels DBpedia

The first test case is focused on the target entity "Mac OS X" (test case 1). As many informatics-related topics, for Mac OS X, the different Webs provide a lot of potentially useful information, although it is not always easy to select the most relevant web resources about a specific subtopic of interest, especially when considering microdata sources such as Twitter. The second test case is related to the entertainment domain and it is about the famous movie saga "Star Wars" (test case 2). Here, we combined web resources taken from two different Linked Data repositories, namely Freebase and DBpedia. Finally, the third test case is focused on the event "London Olympics" (test case 3). In this case, we considered both specialized RSS channels and DBpedia as data sources.

Each student was equipped with the *in*-cloud built by our prototype for each test case over the considered web resources. Moreover, each student was also equipped with a full access to the web, where he/she could access conventional web tools, such as the Freebase web interface, Twitter, and Wikipedia. Each student was then asked to perform a set of search tasks by using first the conventional web tools and then the *in*-cloud. The search tasks proposed were either generic, such as for example "Find useful information about the Mac OS X operating system" or "Find information about

the Olympic Games held in London", or more specific, such as "Find information about critical updating issues concerning Mac OS X" or "Find information about works of fiction inspired by Star Wars". As we want to evaluate also the intuitiveness of *in*-clouds, we gave to the students no further training about *in*-clouds.

As soon as all the users had completed all the search tasks required for a given test case, we collected the user feedback by means of the questionnaire with specific set of questions to be answered for each test case. Moreover, an additional set of questions have been submitted to the users to evaluate the whole experience. In defining the questions, we have followed the following principles [Hoeber and Yang, 2007]: i) collecting subjective reaction after the completion of each set of search tasks provides specific details regarding the participants feelings at each stage in the evaluation; ii) collecting subjective reaction at the conclusion of the whole experience provides an overall view of the participants feelings with respect to the system in general.

According to these principles, the questionnaire was composed by five questions for each test case (see Figure 6.1), plus other five questions concerning the experience as a whole and the comparison of the three test cases (see Figure 6.2). For each question, we use a scale with four-point relevance judgments, capturing varying degrees of user confidence.

Results and discussion. The results presented in Figure 6.1 show a generally positive feedback from the users involved in the experiment, in that the majority of the answers are positive or very positive. For about the 75% of users, conventional web tools do not provide information more relevant than *in*-clouds. However, there are some differences concerning the three experiments. In the test case 1 about "Mac OS X", the user satisfaction was generally slightly lower than the other test cases. This is mainly motivated by two facts. First, conventional web tools provide more information about technological targets, such as "Mac OS X". Thus, it could be easy for users to find relevant information also without *in*-clouds. This is also shown by the percentage of users who claim that some relevant information is missing in the *in*-cloud, which is higher in test case 1 than test case 2 and 3. Moreover, test case 1 involves information taken from Twitter. As a matter of fact, Twitter data quality is lower than Freebase, DBpedia, and RSS channels. As a consequence, the perceived quality of the resulting aggregation in *in*-clouds is also lower than that of test case 2 and 3. Finally, we note that we obtained very good results in test case 3, about "London Olympics". Dealing

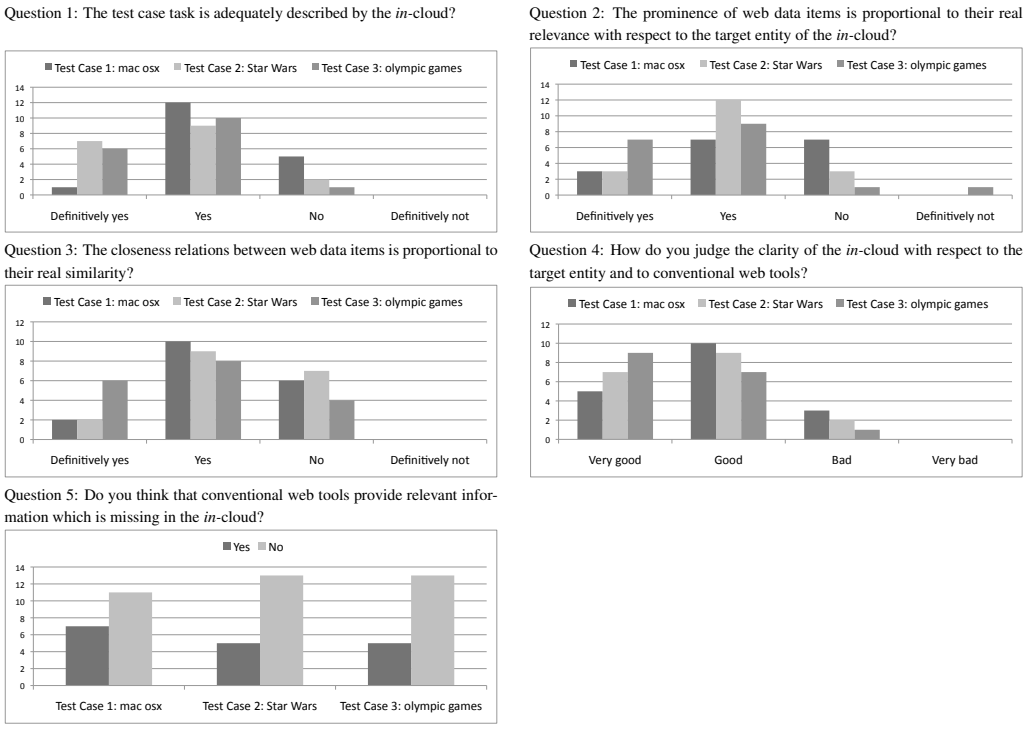
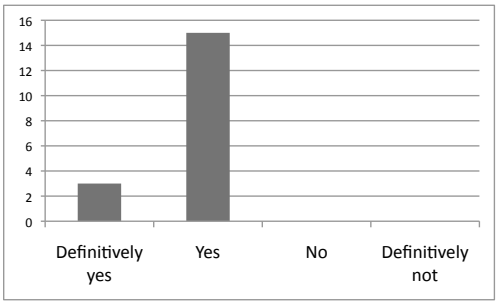


Figure 6.1: Results of user evaluation (test case specific questions)

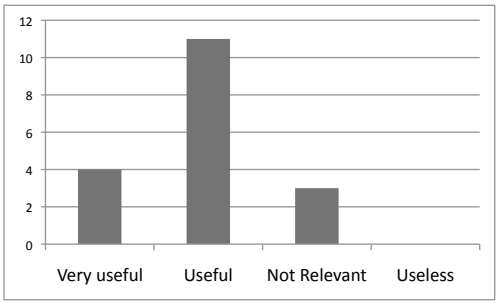
with an event, the advantage of combining together stable information from DBpedia with more fresh information from RSS channels is evident, especially when users are required to execute tasks such as finding an accommodation and looking for the location of a specific sub-event.

From the set of questions concerning the experience as a whole, shown in Figure 6.2, we see how, for the majority of users, *in*-clouds provide an advantage in terms of effectiveness and usability with respect to conventional web search tools. Also the idea of combining together microdata information (from RSS channels and Twitter in our test cases) is considered useful. To this end, it is interesting to note that the test cases where we used microdata (i.e., test cases about "Mac OS X" and "London Olympics") are considered to be more adequately described by *in*-clouds than the test case on "Star Wars", where we just used Linked Data sources. This is an interesting proof that *in*-clouds help users in better accessing web information, especially when multiple kinds of web sources are involved.

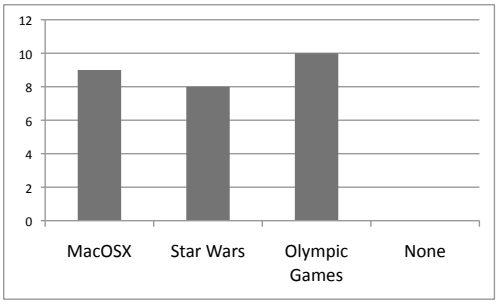
Question 1: Are the *in*-clouds usable with respect to conventional web search tools?



Question 2: How do you judge the usefulness of including RSS and Tweets in the *in*-clouds?



Question 3: Which of the target entities was better described by its corresponding *in*-cloud?



Question 4: How do you judge the effectiveness of conventional web tools and *in*-cloud with respect to the search tasks?

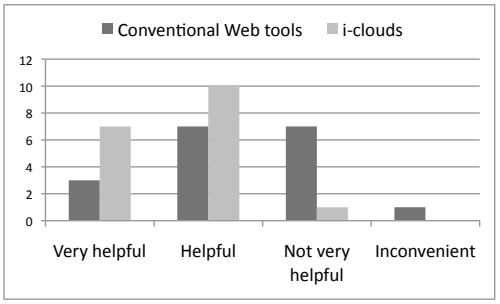


Figure 6.2: Results of user evaluation (general questions)

6.2 System evaluation

Evaluation issues concerning the system evaluation are focused on three kinds of tests. In the first set of tests, we have the goal of evaluating the accuracy of *in*-clouds, that is the capability of our semantic clouding approach to collect in the same *in*-cloud web data items really prominent with respect to a given target entity. In the second set of tests, we evaluate the cohesion of *in*-clouds, that is the impact on the *in*-cloud construction of the closeness threshold for candidate clusters selection. In the third set of tests, we evaluate the scalability of the proposed approach. All the tests have been run against the experimental data collection described in Chapter 3 and used also for the examples in this paper.

6.2.1 Accuracy evaluation

Hypothesis. In this section, we evaluate the *in*-cloud accuracy, which can be measured in terms of the accuracy of the matching techniques that are employed for calculating the similarity between web data items. In our approach, matching is executed by exploiting our matching tool HMatch 2.0 [Castano et al., 2008, 2010a] where we implemented the various matching techniques discussed in Chapter 4.

Experimental setup. For evaluating the quality of HMatch 2.0 we rely on the IIMB 2010 dataset[1] (see Appendix A) and related tools that have been used for the International Instance Matching Evaluation Contest of the Ontology Alignment Evaluation Initiative (OAEI)[2]. As explained in details in Appendix A, IIMB provides a benchmark which supports the controlled generation of data modifications and expected results for a variety of data. In particular, a given set of data is artificially changed in several ways by introducing various kinds of value, structure, and logic modifications. Then, a matching tool is evaluated according to its capability of retrieving a correspondence between a data item and its modified counterpart, that is the expected correspondence. For accuracy evaluation, IIMB has been exploited to generate specific groups of tests for web data items related to the web resources involved in our three test cases (see Section 6.1).

Results and discussion. The accuracy of our matching techniques has been evaluated using the FMeasure, the conventional accuracy indicator defined as the harmonic mean of *precision* and *recall*[3]. The results of accuracy evaluation are shown in Figure 6.3.

HMatch 2.0 has been also compared against a simple matching algorithm called StringMatch that just performs matching using different conventional string matching techniques and selecting the best result. This kind of comparison is a baseline benchmark when the matching process mainly involves strings. The goal of the comparison

[1] http://www.instancematching.org/oaei/imei2010.html

[2] http://oaei.ontologymatching.org/2010

[3] Precision is proportional to the number of expected correspondences among those retrieved by the matching tool. Recall is proportional to the number of expected correspondences that are retrieved by the matching tool.

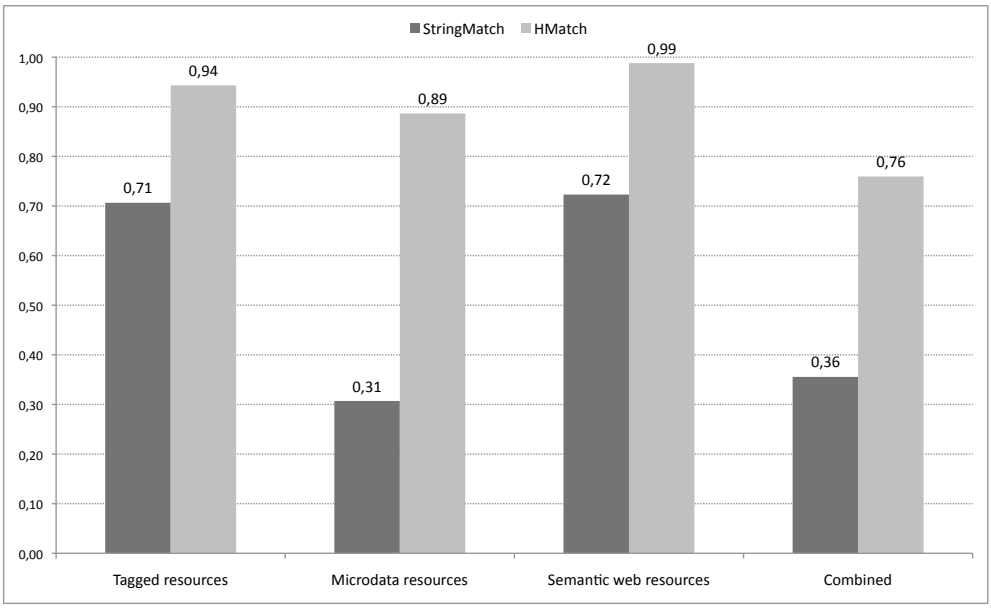


Figure 6.3: Accuracy of the matching techniques measured with the FMeasure

is to show the advantages of using closeness coefficients instead of more conventional string matching values for the web data item classification. These results show how the accuracy obtained using the HMatch 2.0 closeness measures is significantly better than the one obtained using StringMatch in all the test cases, and especially dealing with microdata resources that require to work on the meaning and social nature of the string that are compared.

A further accuracy test concerns the relation between precision and recall, as shown in Figure 6.4.

Usually, precision is higher when recall is lower and vice-versa. In the literature, it has been observed that, if we analyze the curve representing the relation between precision and recall, strong algorithms are featured by a convex curve, while weak algorithms are featured by a concave curve [Euzenat and Shvaiko, 2007]. In our experiments, we see how HMatch 2.0 is stronger than StringMatch, especially when we look for balanced results in terms of precision and recall.

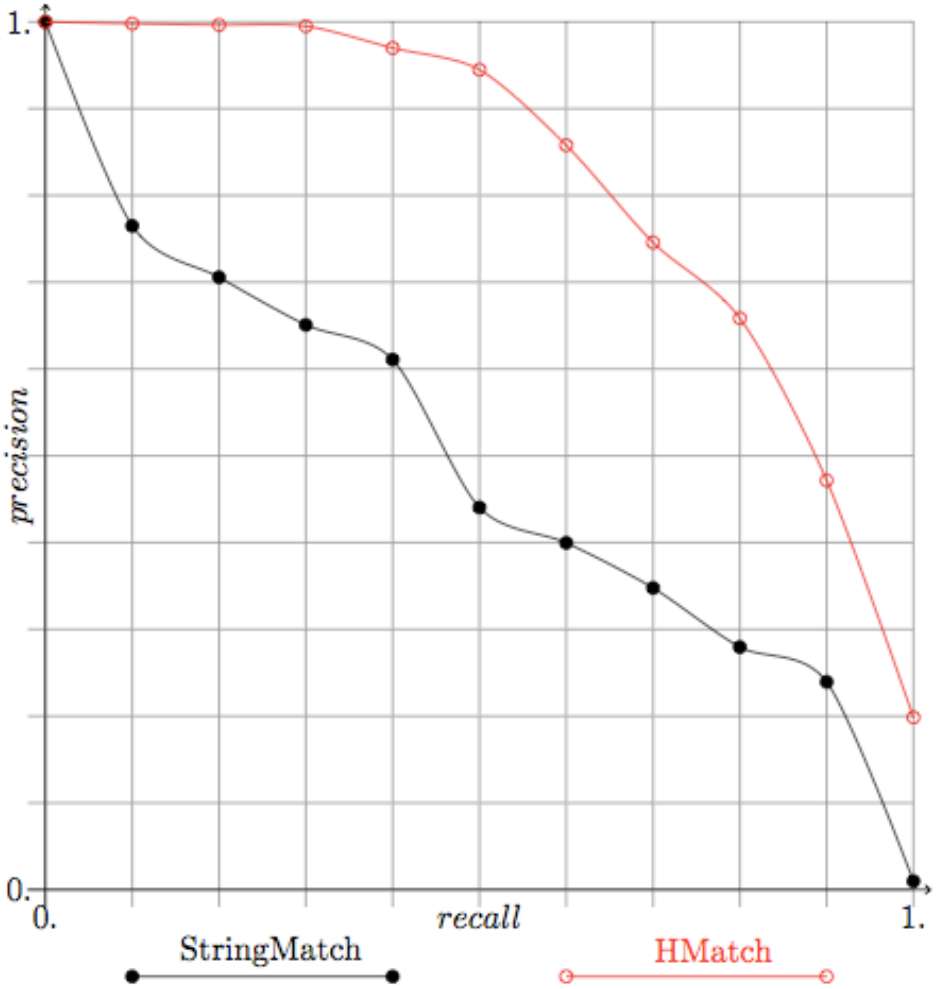


Figure 6.4: Relation between precision and recall of StringMatch and HMatch 2.0

6.2.2 Cohesion evaluation

Hypothesis. In this section, we evaluate the *in*-cloud cohesion on the basis of the closeness threshold (see Section 5.2) that is chosen for selecting the candidate clusters. The cohesion of an *in*-cloud can be defined as the average closeness between the web data items therein contained. By setting a high threshold, the resulting *in*-cloud will have a small dimension and a high cohesion; by setting a low threshold, the resulting *in*-cloud will have a high dimension and a low cohesion. The objective of this set

of experiments is to identify a tradeoff between the dimension of an *in*-cloud and its cohesion.

Experimental setup. In order to evaluate the impact of the closeness threshold on the *in*-cloud cohesion, we have constructed an *in*-cloud for each main character in the Star Wars saga. Then, we have measured the average number of web data items per *in*-cloud and the average level of closeness between web data items in the *in*-clouds according to different values of the closeness threshold.

Results and discussion. In Figure 6.5, we report the result of this set of experiments by normalizing the number of web data items per *in*-cloud with respect to the total number of web data items in the largest *in*-cloud, that is the one obtained using a closeness threshold equal to 0.1.

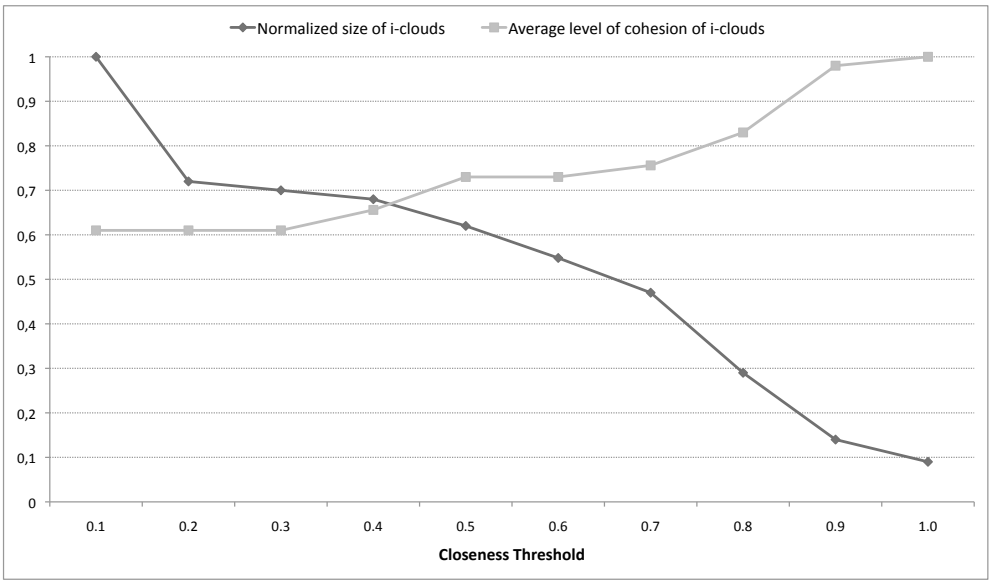


Figure 6.5: Normalized size and average level of cohesion of *in*-clouds with respect to the closeness threshold

As expected, incrementing the value of the closeness threshold, the size of *in*-clouds decreases while the average value of their cohesion increases. This observation can be useful for the selection of the default closeness threshold value. In fact, a closeness

threshold of 0.5 is shown to be enough to guarantee a good level of cohesion (i.e., *in*-clouds with average closeness higher than 0.7) while keeping the size of the *in*-cloud still significant (i.e., not trivially limited to the few web data items exactly matching the target entity).

6.2.3 Scalability evaluation

Hypothesis. In this section, we evaluate the scalability of our semantic clouding approach, in terms of computation time, as the number of considered web data items grows. In particular, we observe that scalability is mainly affected by clustering techniques, although there are many well known and standard techniques to reduce the number of matching operations in case of large data collections [Euzenat and Shvaiko, 2007]. Thus, in this set of experiments, we evalutate the scalability of the hierarchical and agglomerative clustering procedure described in Section 5.1.

Experimental setup. In order to evaluate the scalability of the clustering procedure, we considered a growing number of web data items, ranging from 30 up to 5000.

Results and discussion. The results of the scalability evaluation are shown in Figure 6.6.

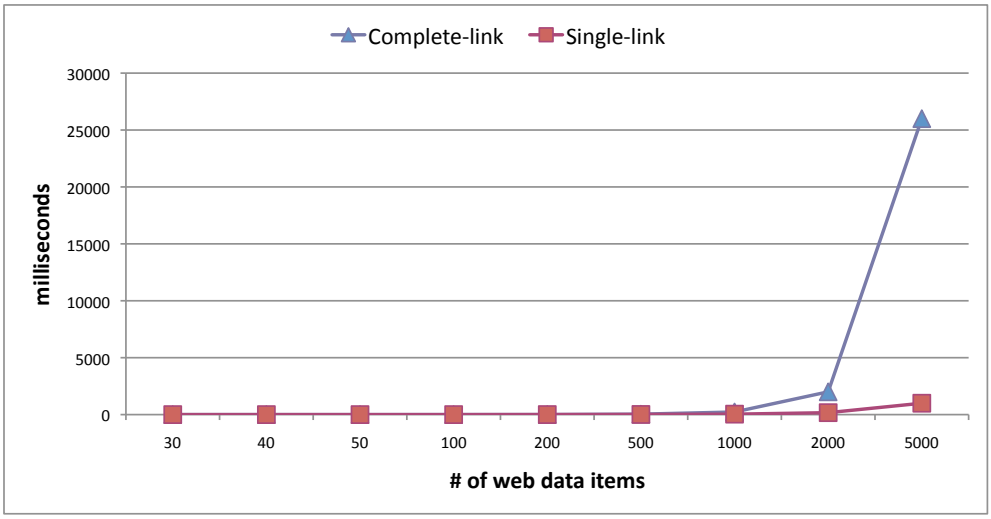


Figure 6.6: Scalability of the clustering procedure

Time complexity of hierarchical and agglomerative clustering is $O(n^2)$, where n is the number of web data items in the WDI repository. However, the results in Figure 6.6 show that the approach scales well if the single-link clustering is chosen. In fact, single-link clustering is useful to reduce the number of clusters, since a lower level of closeness is required between two clusters to be merged. On the contrary, the complete-link clustering creates a high number of different clusters, since it requires a higher level of closeness between two clusters to be merged. However, in case of very large collections of web data items, or in order to use complete-link clustering techniques, the scalability issues may be addressed by performing the classification of web resources (i.e., the matching and clustering operations) off-line, in a batch manner, as will be discussed in Chapter 7.

Chapter 7

Conclusions and future work

In this thesis, we presented our approach to web resource clouding for the construction of cross-web, disciplined, and intuitive *in*-clouds of prominent information about a given target entity. A Java-based prototype for *in*-cloud construction has been developed and it is available to the public[1]. The presented matching techniques have been included in the HMatch 2.0 environment[2]. The positive results we obtained during evaluation encourage to continue working on *in*-cloud research issues. In particular, we are working on defining a set of operations between *in*-clouds (e.g., selection, projection, join). Furthermore, a focused search application based on *in*-clouds is being developed in the domain of tourism and entertainment related to the city of Milan.

More specifically, for what concerns the term matching techniques, goals of future work regard the capability to automatically identify semantic relations between the different term-components of compound words and to manage the term similarity between non-English words. In particular, the semantic relations between the term-components of compound words can be identified by manually analyzing the set of recurrent composition patterns of common compound words, and by defining heuristics to automatically identify the most important component of each compound word. Non-English terms can be managed by exploiting different external sources, such as multi-language dictionaries and/or web-based encyclopedias like Wikipedia or special-purpose dictionaries.

[1] http://islab.dico.unimi.it/clouding

[2] http://islab.dico.unimi.it/hmatch

Finally, we are also working on the definition and the evaluation of two different application scenarios for semantic clouding, namely the *clouding in-the-large* and the *clouding in-the-small*, in order to manage the scalability issues discussed in Chapter 6.

Clouding in-the-large. This is the scenario of domain-independent search applications, like for example general-purpose search engines, where the web resources that populate the WDI repository are acquired from all the Webs without any filtering operation. Due to the potentially huge number of web resources involved (e.g., many millions of web data items in the WDI repository), matching of web data items and definition of the closeness tree are executed off-line, in a batch manner. The subsequent phase of *in*-cloud construction is interactively performed over the existing closeness tree, upon specification of the target entity of interest by the requesting user. Periodically, the acquisition of new web resources and the refresh of the previously stored ones are executed to update the WDI repository and to enrich it with new web data items. This step is particularly important for tagged and microdata resources that have a rapid obsolescence and need to be frequently refreshed for being up-to-date with respect to the very last user comments. The update of the WDI repository requires the update of the closeness tree as well. This can be performed incrementally when most of the updates involves the insertion of new web data items. Instead, a "from scratch" reconstruction of the closeness tree can be the preferable solution when the update of the WDI repository involves the refresh of many web data item already stored in the WDI repository. Usually, in both these solutions, a long amount of batch work is required. For this reason, the update of the closeness tree is performed periodically, once that a sufficient number of new web resources has been accumulated. In general, *in*-clouds are generated from the currently available closeness tree, and they do not take into account the web data items just inserted in the WDI repository. A caching mechanism can be adopted to reduce the response time between the specification of the target entity and the visualization of the resulting *in*-cloud. Caching can be managed through a history-based criterion, where the last k-queried *in*-clouds are maintained. Alternatively, a popularity-based criterion can be used, where the cached *in*-clouds are those that are most frequently queried by users. According to the periodic enrichment of the WDI repository and to the subsequent closeness tree update, the cached *in*-clouds will be updated as well.

Clouding in-the-small. This is the scenario of domain-specific search applications, like for example focused search engines, where selected portions of the Webs are used to populate the WDI repository according to a predefined set of filtering operations. For instance, domain-based and context-based criteria can be used to determine the portion(s) of the Webs to acquire in the WDI repository. In the domain-based criterion, the filtering operations consist in populating the WDI repository with the web resources that match a given domain expressed by a predefined list of keywords. In the context-based criterion, a context model is provided to specify the constraints that a web resource needs to satisfy for being acquired in the WDI repository. For example, a context can specify that only the web resources about a certain geographic location (geographic constraint) and published in a specific period of time (temporal constraint) have to be acquired in the WDI repository. According to the scale of the WDI repository, matching of web data items and closeness tree definition can be executed either off-line or on-line. As for clouding in-the-large, the off-line option is preferable when the scale of the WDI repository is too much high for allowing the closeness tree definition in an acceptable waiting time for the requesting user (e.g., about one million of web data items in the WDI repository). The on-line option is preferable with very focused WDI repositories (e.g., less than one million of web data items in the WDI repository). In this case, both closeness tree definition and *in*-cloud construction are performed on-the-fly upon specification of the target entity by the requesting user. The advantage of the on-line option is that the resulting *in*-clouds are up-to-date with respect to the WDI repository, since refresh and periodic enrichment of the stored web data items are immediately considered.

Appendix A

Benchmark for matching techniques evaluation

In this appendix, we describe the benchmark [Ferrara et al., 2008] that has been used to evaluate the matching techniques that have been described in this thesis. In particular, such benchmark has been developed to test instance matching algorithms working on ontology instances, that is, Semantic Web resources. To this end, different kinds of modifications (i.e., data value modifications, structural modifications, and logical modifications) have been applied on an original set of ontology instances, in order to test if an instance matching algorithm is able to properly identify the corresponding modified counterpart of each original instance. However, the benchmark can be used as well to evaluate matching algorithms working only on tagged and microdata resources, by considering, respectively, only the data value modifications, or only the data value and the structural modifications. Finally, by submitting to the matching algorithm a set of instances which have been modified in different ways, the benchmark can be used to test the results obtained by the algorithm in case of comparing heterogeneous kinds of web resources (e.g., tagged, microdata, and Semantic Web resources).

A.1 Design of the benchmark

A widely recognized problem in the Semantic Web is the lack of evaluation data. While OAEI[1] (Ontology Alignment Evaluation Initiative) has provided a reference bench-

[1] http://oaei.ontologymatching.org/2007/benchmarks

mark for schema matching [Euzenat and Shvaiko, 2007], evaluation data for instance matching are still few. In [Christen, 2008b,c], the author presents a system, Febrl, which allows researchers to compare a new record linkage algorithm with all the main record linkage techniques. In [Chandel et al., 2007], the authors describe the construction of a benchmark dataset for data cleaning approaches, obtained by setting the distribution of duplicates, the percentage of erroneous duplicates, and the extent of error in each erroneous duplicate. Further works dealing with schema matching evaluation are [Hollink et al., 2008; Isaac et al., 2008].

The aim of our benchmark is to provide a complete set of tests for instance matching algorithms evaluation. In particular, we do not only define a specific benchmark, but also we provide an architecture for the definition of a semi-automatic procedure for the generation of several different benchmarks. In Figure A.1, the overall process of benchmarks generation is shown.

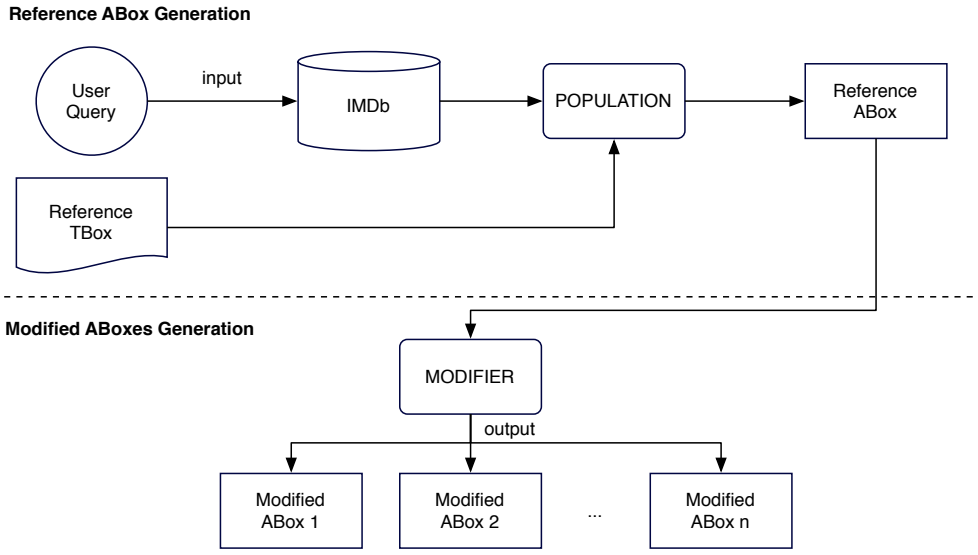


Figure A.1: Benchmarks generation

As an example of this general procedure, we describe in the following a specific instantiation of it, that is the creation of a specific benchmark for instance matching[2].

[2] http://islab.dico.unimi.it/iimb

Reference ABox generation. First of all, we chose a domain of interest (i.e., the domain of movie data), and we created a reference ($\mathcal{ALCF}(\mathcal{D})$) TBox for it, based on our knowledge of the domain[3]. This contains 15 named classes, 5 object properties and 13 datatype properties. The reference TBox is then populated by automatically creating a reference ABox. Data are extracted from IMDb[4] by executing a query Q of the form:

$$SELECT * FROM\ movies\ WHERE\ title\ LIKE\ '\%X\%'$$

where X is a variable specifying a word of our choice. Thus, all selected movies contain the word X in their title. The corresponding individuals in the reference ABox are referred to similar objects, but each of them represents a distinct real-world object. As a consequence, each instance can be univocally identified.

In order to get our reference ABox, we put $X = Scarface$. The reference ABox obtained in that way contains 302 individuals, that is all the movie objects matching the query and all the actors in the movie cast.

Modified ABoxes generation. Once the reference ABox is created, we generate a set of modified ABoxes, each consisting in a collection of instances obtained modifying the corresponding instances in the reference ABox. Transformations introduced in benchmark ABoxes can be distinguished into three main categories. Modifications belonging to different categories are also combined together within the same ABox.

A.2 Generating instance modifications

In this section, we describe the *Modifier* module of our benchmarks generation procedure, that is the way the modified ABoxes of benchmarks are generated. Given the reference ABox as input, and a user specification of all the transformations to apply on it, the *Modifier* module automatically produces the corresponding modified ABoxes. In the following, all the modifications that can be applied on the reference ABox are presented.

[3] http://islab.dico.unimi.it/ontologies/benchmark/imdbT.owl

[4] http://www.imdb.com

A.2.1 Data value modifications

The goal of this first category of modifications is to simulate the differences that can be found between instances referred to the same real-world object at the property value level. Those include typographical errors, use of different standard formats to represent the same value, or a combination of both within the same value.

Typographical errors. Real data are often dirty. That is mainly due to typographical errors made by humans while describing data. In order to simulate typographical errors, we use a function that takes as input a datatype property value and produces as output a modified value. This kind of transformation can be applied to each datatype property value (e.g., string value, integer value, date value). The modifications to apply on the input value are randomly chosen between the following:

- *Insert character.* A random character (or a random number, if the property has a numerical value) is inserted in the input value at a random position.
- *Modify character.* A random character (or a random number, if the property has a numerical value) is modified in the input value.
- *Delete character.* A random character (or a random number, if the property has a numerical value) is deleted in the input value.
- *Exchange characters position.* The position of two adjacent characters (or two adjacent numbers, if the property has a numerical value) is exchanged in the input value.

For example, the movie title “Scarface” can be transformed into the modified value “Scrface”, obtained deleting a random character from the original string.

In addition, it is possible to specify the level of severity (i.e., low, medium or high) in applying such transformations. Anyway, the number of transformations introduced in the input value is proportional to the value’s length. If the number of transformations to apply is greater than one, the corresponding value can be modified combining different transformations.

Typographical modifications can be applied to "identifying properties", "non-identifying properties" or both. That classification is based on the analysis of the percentage of null and distinct values specified for the selected property. In particular, properties with a high percentage of distinct values and a low percentage of null values are classified as the most identifying.

Of course, the total amount of modifications applied to each modified ABox has to change the reference ABox in a way that it is still reasonable to consider the two ABoxes semantically equivalent. In other words, a modified ABox is included in the benchmark only if a human can understand that its instances are referred to the same real-world object, as the ones belonging to the reference ABox. Thus, in order to evaluate the distance between the reference ABox and each modified ABox, we introduce a measure that takes into account the number of modifications applied to the same ABox, the kind of the properties (i.e., "identifying properties" or "non-identifying properties") which have been modified, and the level of severity of the modifications (i.e., low, medium or high). However, this measure does not affect the instance matching results in a deterministic way, since they depend on the weight that the tested algorithm gives to each kind of modification. Anyway, we assume that a modified ABox can be considered semantically equivalent to the reference ABox only if it changes no more than 20% of each instance description.

Use of different standard formats. The same data within different sources can be represented in different ways. In order to simulate the use of different standards within different sources, we use a function that takes as input a property value which allows standard modifications (e.g., person name) and produces as output a modified value, using a different standard format. For example, the director name "De Palma, Brian" can be transformed in the modified value "Brian De Palma", which is another standard format to specify a person name.

A.2.2 Structural modifications

Another kind of situation that is simulated in our instance matching benchmark is the comparison between instances with different structures. In fact, the same individual feature (i.e., each instance property) can be modeled in different ways. Moreover, dif-

ferent descriptions of the same real-world object can specify different subsets, eventually empty, of all the possible values for that property. Combinations of different transformations belonging to this class of modification are also applied in the benchmark.

Use of different levels of depth for properties representation. A first example of this class of modifications is shown in Figure A.2.

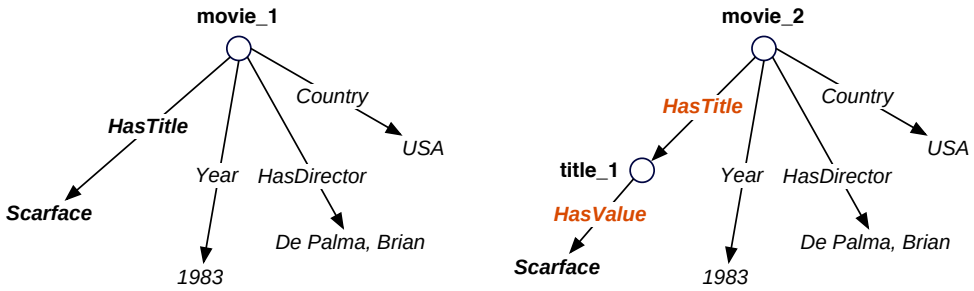


Figure A.2: Use of different levels of depth to represent the same property

The two instances *movie_1* and *movie_2* are both referred to the same film, but the movie title property is modeled in two different ways. In fact, the title of *movie_1* is specified directly through a datatype property value, while the title of *movie_2* is specified through a reference to another individual which has a property with the same title value (i.e., "Scarface"). In particular, in the first representation, the property $HasTitle$ is a datatype property, while in the second one it is an object property and its value is the reference to $title\_1$ instance.

In order to simulate the comparison between instances with different structures, we use a function that takes as input a datatype property and produces as output an object property with the same name. Moreover, the function creates a new attribute to the generated object property, whose value is the same as the original datatype property.

Use of different aggregation criteria for properties representation. In an analogous way, the name of a person can be stored all within the same property, or it can be split into different properties such as, for example, *Name* and *Surname*. Figure A.3 shows two different ways of modeling the name "Pacino, Al".

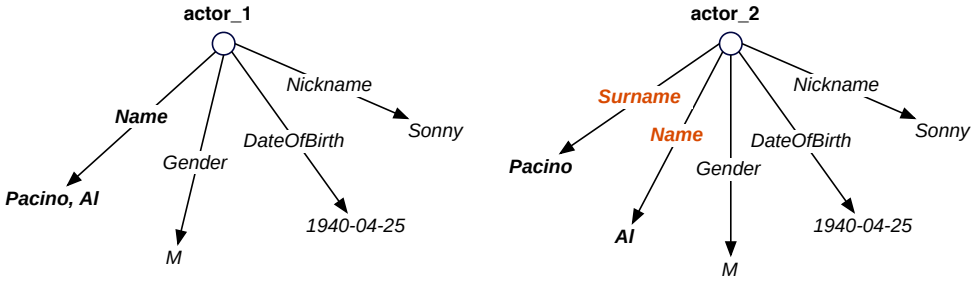


Figure A.3: Use of different aggregation criteria to represent the same property

In the first representation the whole value is stored within the property *Name*, while in the second one the string is split into the two values "Pacino" and "Al", referred to the properties *Name* and *Surname*, respectively.

In order to simulate the comparison between properties modeled in different ways, we use a function that takes as input a datatype property value that can be split and produces as output two new datatype properties, each specifying a different part of the original value.

Missing values specification. A further example of structural heterogeneity is shown in Figure A.4.

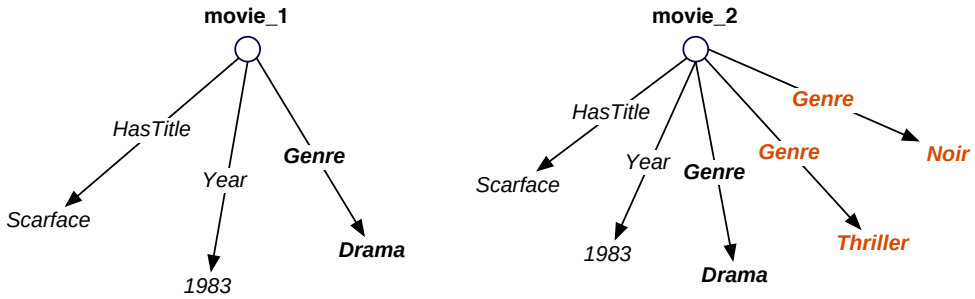


Figure A.4: Specification of different subsets of values on the same multi-values property

The two instances *movie_1* and *movie_2* are both referred to the same film, but the

two different descriptions specify different subsets of values on the property *Genre*.

In order to simulate the comparison between different sets of values referred to the same property, we use a function that takes as input the set of values specified for a selected property and produces as output a subset, eventually empty, of it. This kind of transformation can be applied to each property. Moreover, if a property allows multiple values, it is possible to specify if deleting all the values of the selected property or a random number of them.

A.2.3 Logical modifications

Finally, the instance matching process should take into account the need to use some kind of reasoning, in order to correctly find instances to be compared. In fact, instances referring to the same real-world object can be instantiated in different ways within different ontologies. In the following, we describe five kinds of situations that we develop in our benchmark, that can also be combined together. Each situation requires some kind of reasoning. Examples of those are shown in Figure A.5.

Reference TBox

$Movie$	$\sqsubseteq$	$Item$	$Movie \sqcap Product$	$\sqsubseteq$	$\bot$
$Film$	$\sqsubseteq$	$Item$	$Movie$	$\equiv$	$\forall p.G$
$Product$	$\sqsubseteq$	$Item$	$SubM$	$\equiv$	$\forall p.SubG$
$Action$	$\sqsubseteq$	$Movie$	$SubG$	$\sqsubseteq$	G

Reference ABox

$movie\_1$	:	$Movie$
$movie\_2$	:	$Movie$
$movie\_3$	:	$Movie$
$movie\_4$	:	$Movie$
$movie\_5$	:	$Movie$
$(movie\_5, \text{“}Scarface\text{”})$	:	$HasTitle$

Modified ABox

$movie\_1$	:	$Film$
$movie\_2$	:	$Product$
$movie\_3$	:	$Action$
$movie\_4$	:	$SubM$
$movie\_5$	:	$Movie$
$movie\_5$	:	$\exists HasTitle.\text{“}Scarface\text{”}$

Figure A.5: Example of logical modification

Instantiation on different subclasses of the same superclass. This modification is obtained instantiating identical individuals into different subclasses of the same class. For example, in our benchmark, all the movie objects are instances of class *Movie* in the reference ABox. Instead, in one of the modified ABoxes, we change the type of those individuals, making them instances of class *Film*. Classes *Movie* and *Film* are both subclasses of *Item*. In Figure A.5, *movie_1* is instance of *Movie* in the reference ABox, while it is instance of *Film* in the modified ABox. Instance matching algorithms are thus required to recognize that those two instances are referred to the same real-world object, even if they belong to different concepts.

Instantiation on disjoint classes. This modification is obtained instantiating identical individuals into disjoint classes. For example, in one of the modified ABoxes, we change the type of all the instances of the class *Movie*, making them instances of class *Product*. Classes *Movie* and *Product* are defined as disjoint classes in the reference TBox. In Figure A.5, *movie_2* is instance of *Movie* in the reference ABox, while it is instance of *Product* in the modified ABox. In this case, the tested algorithms has to be able to recognize that instances belonging to disjoint classes cannot be referred to the same real-world object, even if they seem identical.

Instantiation on different classes of a class hierarchy explicitly declared. This modification is obtained instantiating identical individuals into different classes on which an explicit class hierarchy is defined. For example, an individual representing a movie can be classified as an instance of the general concept *Movie*, as it is in the reference ABox, or it can be classified as an instance of a more specific subclass of it, such as *Action*, *Biography*, *Comedy* or *Drama*, depending on the value that the movie instances specify on the property *Genre*. In Figure A.5, *movie_3* is instance of *Movie* in the reference ABox, while it is instance of its subclass *Action* in the modified ABox, since it is an action movie. Instance matching algorithms are thus required to recognize that those two instances are referred to the same real-world object, even if they belong to different concepts within the class hierarchy. This explicit class hierarchy declaration can be recognized using a RDFS reasoner.

Instantiation on different classes of a class hierarchy implicitly declared. A further modification that we apply in the benchmark is the instantiation of identical indi-

viduals into different classes on which an implicit class hierarchy is defined. Such an implicit class hierarchy declaration can be obtained through the use of restrictions. For example, the restrictions specified on classes *Movie* and *SubM* in the reference TBox, implicitly declare that *SubM* is a subclass of *Movie*. In Figure A.5, *movie_4* is instance of *Movie* in the reference ABox, while it is instance of *SubM* in the modified ABox. Instance matching algorithms are thus required to recognize that those two instances are referred to the same real-world object, even if they belong to different concepts which are not explicitly related. This implicit class hierarchy declaration can be recognized using a DL reasoner.

Implicit values specification. Another use of restrictions that requires a reasoning process, is the comparison between an explicit specified value and an implicit specified one, that is using an *hasValue* restriction. This kind of situation is simulated in our benchmark by adding a new type for each instance of the modified ABox. This type is a class that (implicitly) specifies property values through an *hasValue* restriction. In Figure A.5, in the reference ABox, *movie_5* is instance of *Movie* and its value on the property *HasTitle* is "Scarface"; in the modified ABox, *movie_5* is as well instance of *Movie*, but it is also instance of the restriction class that implicitly specifies the value "Scarface" for its *HasTitle* property. Instance matching algorithms are thus required to recognize that those two instances are referred to the same real-world object, even if some property values of the modified instances are implicitly defined.

Bibliography

B. Aleman-Meza, C. Halaschek-Wiener, I.B. Arpinar, and A.P. Sheth. Context-Aware Semantic Association Ranking. In *Proc. of the 1st International Workshop on Semantic Web and Databases (SWDB 2003) co-located with the 29th International Conference on Very Large Databases (VLDB 2003)*, Berlin, Germany, 2003.

K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao. Describing Linked Datasets. In *Proc. of the WWW International Workshop on Linked Data on the Web (LDOW 2009)*, Madrid, Spain, 2009.

S. Auer, R. Doehring, and S. Dietzold. LESS - Template-Based Syndication and Presentation of Linked Data. In *Proc. of the 7th Extended Semantic Web Conference (ESWC 2010)*, Heraklion, Greece, 2010.

R.A. Baeza-Yates and B.A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press/Addison-Wesley, 1999.

M. Barla and M. Bieliková. On Deriving Tagsonomies: Keyword Relations Coming from Crowd. In *Proc. of the 1st International Conference on Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*, Berlin, Heidelberg, 2009.

G. Begelman, P. Keller, and F. Smadja. Automated Tag Clustering: Improving Search and Exploration in the Tag Space. In *Proc. of the Collaborative Web Tagging Workshop at the WWW 2006*, Edinburgh, Scotland, 2006.

S. Bergamaschi, F. Guerra, M. Orsini, C. Sartori, and M. Vincini. RELEVANT News: a Semantic News Feed Aggregator. In *Proc. of the 4th Workshop on Semantic Web Applications and Perspectives (SWAP 2007)*, Bari, Italy, 2007.

T. Berners-Lee, R. Cailliau, A. Luotonen, H.F. Nielsen, and A. Secret. The World-Wide Web. *Communications of the ACM*, 37(8), 1994.

T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5), 2001.

T. Berners-Lee, J. Hollenbach, K. Lu, J. Presbrey, E. Prud'hommeaux, and M. Schraefel. Tabulator Redux: Browsing and Writing Linked Data. In *Proc. of the WWW International Workshop on Linked Data on the Web (LDOW 2008)*, Beijing, China, 2008.

I. Bhattacharya and L. Getoor. Iterative Record Linkage for Cleaning and Integration. In *Proc. of the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD 2004)*, New York, NY, USA, 2004.

C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3), 2009.

P. Bouquet, H. Stoermer, M. Mancioppi, and D. Giacomuzzi. OkkaM: Towards a Solution to the "Identity Crisis" on the Semantic Web. In *Proc. of the 3rd Italian Semantic Web Workshop*, Pisa, Italy, 2006.

P. Bouquet, H. Stoermer, C. Niederee, and A. Mana. Entity Name System: The Backbone of an Open and Scalable Web of Data. In *Proc. of the IEEE International Conference on Semantic Computing (ICSC 2008)*, 2008.

S. Castano, V. De Antonellis, and S. De Capitani Di Vimercati. Global Viewing of Heterogeneous Data Sources. *IEEE Transactions on Knowledge and Data Engineering*, 13(2), 2001.

S. Castano, S. Espinosa, A. Ferrara, V. Karkaletsis, A. Kaya, S. Melzer, R. Möller, S. Montanelli, and G. Petasis. Multimedia Interpretation for Dynamic Ontology Evolution. *Journal of Logic and Computation*, 2008.

S. Castano, A. Ferrara, V. Karkaletsis, D. Lorusso, S. Montanelli, G. Petasis, P. Sanna, and G. Varese. Ontology Evolution Toolkit: Final Version. Technical report, D4.12. BOEMIE: Bootstrapping Ontology Evolution with Multimedia Information Extraction, IST Project n. FP6-027538, 6th EU Framework Programme, 2009a.

S. Castano, A. Ferrara, and S. Montanelli. Matching Ontologies in Open Networked Systems: Techniques and Applications. *Journal on Data Semantics (JoDS)*, 5, 2006.

S. Castano, A. Ferrara, and S. Montanelli. Dealing with Matching Variability of Semantic Web Data Using Contexts. In *Proc. of the 22nd International Conference on Advanced Information Systems Engineering (CAiSE 2010)*, Hammamet, Tunisia, 2010a.

S. Castano, A. Ferrara, S. Montanelli, and G. Varese. Matching Semantic Web Resources. In *Proc. of the 8th International Workshop on Web Semantics (WebS 2009) co-located with the 20th International Conference on Database and Expert Systems Applications (DEXA 2009)*, Linz, Austria, 2009b.

S. Castano, A. Ferrara, S. Montanelli, and G. Varese. Semantic Coordination of P2P Collective Intelligence. In *Proc. of the International ACM Conference on Management of Emergent Digital EcoSystems (MEDES 2009)*, Lyon, France, 2009c.

S. Castano, A. Ferrara, S. Montanelli, and G. Varese. A Semantic Clouding Approach for Cross-Webs Interoperability. In *Proc. of the 3rd Interop-Vlab.it Workshop co-located with the 7th Conference of the Italian Chapter of AIS (itAIS 2010)*, Naples, Italy, 2010b.

S. Castano, A. Ferrara, S. Montanelli, and G. Varese. Matching Micro-Data. In *Proc. of the 18th Italian Symposium on Advanced Database Systems (SEBD 2010)*, Rimini, Italia, 2010c.

S. Castano, A. Ferrara, S. Montanelli, and G. Varese. Similarity-based Classification of Microdata. In A. D'Atri, M. Ferrara, J.F. George, and P. Spagnoletti, editors, *Information Technology and Innovation Trends in Organizations*. Springer-Verlag, 2010d.

S. Castano, A. Ferrara, S. Montanelli, and G. Varese. Knowledge-Driven Multimedia Information Extraction and Ontology Evolution. In G. Paliouras, C.D. Spyropoulos, and G. Tsatsaronis, editors, *Ontology and Instance matching*. Springer-Verlag, 2011.

C. Cattuto, A. Baldassarri, V.D.P. Servedio, and V. Loreto. Emergent Community Structure in Social Tagging Systems. *Advances in Complex Systems*, 11(4), 2008.

C. Cattuto, C. Schmitz, A. Baldassarri, V.D.P. Servedio, V. Loreto, A. Hotho, M. Grahl, and G. Stum. Network Properties of Folksonomies. *AI Communications*, 20(4), 2007.

A. Chandel, O. Hassanzadeh, N. Koudas, M. Sadoghi, and D. Srivastava. Benchmarking Declarative Approximate Selection Predicates. In *Proc. of the 5th ACM SIGMOD International Conference on Management of Data (SIGMOD 2007)*, New York, NY, USA, 2007.

M. Chen, J.T. Sun, H.J. Zeng, and K.Y. Lam. A Practical System of Keyphrase Extraction for Web Pages. In *Proc. of the 14th ACM Int. Conference on Information and Knowledge Management (CIKM 2005)*, Bremen, Germany, 2005.

E.H. Chi. The Social Web: Research and Opportunities. *Computer IEEE Computer Society*, 41(9), 2008.

P. Christen. A Two-Step Classification Approach to Unsupervised Record Linkage. In *Proc. of the 6th Australasian Data Mining Conference (AusDM 2007)*, Gold Coast, Australia, 2007.

P. Christen. Automatic Record Linkage using Seeded Nearest Neighbour and Support Vector Machine Classification. In *Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2008a.

P. Christen. Febrl - A Freely Available Record Linkage System with a Graphical User Interface. In *Proc. of the 2nd Australasian Workshop on Health Data and Knowledge Management*, Darlinghurst, Australia, 2008b.

P. Christen. Febrl - An Open Source Data Cleaning, Deduplication and Record Linkage System with a Graphical User Interface. In *Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2008c.

L. Deligiannidis, K.J. Kochut, and A.P. Sheth. RDF Data Exploration and Visualization. In *Proc. of the ACM 1st Workshop on CyberInfrastructure: Information Management in eScience*, New York, NY, USA, 2007.

D. Dey, S. Sarkar, and P. De. Entity Matching in Heterogeneous Databases: A Distance Based Decision Model. In *Proc. of the 31th Annual Hawaii International Conference on System Sciences (HICSS 1998)*, Kohala Coast, Hawaii, USA, 1998.

D. Dey, S. Sarkar, and P. De. A Distance-Based Approach to Entity Reconciliation in Heterogeneous Databases. *IEEE Transactions on Knowledge and Data Engineering*, 14(3), 2002.

D.M. Dunlavy, J.P. Conroy, and D.P. O'Leary. QCS: A Tool for Querying, Clustering, and Summarizing Documents. In *Proc. of the Human Language Technology Conference (NAACL 2003)*, 2003.

D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

F. Echarte, J.J. Astrain, A. Córdoba, and J. Villadangos. Ontology of Folksonomy: A New Modeling Method. In *Proc. of Semantic Authoring, Annotation and Knowledge Markup (SAAKM 2007)*, 2007.

T. Eda, M. Yoshikawa, T. Uchiyama, and T. Uchiyama. The Effectiveness of Latent Semantic Analysis for Building Up a Bottom-up Taxonomy from Folksonomy Tags. *World Wide Web*, 12(4), 2009.

J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer-Verlag, 2007.

I.P. Fellegi and A.B. Sunter. A Theory for Record Linkage. *Journal of the American Statistical Association*, 64, 1969.

A. Ferrara, D. Lorusso, S. Montanelli, and G. Varese. Towards a Benchmark for Instance Matching. In *Proc. of the 3rd International Workshop on Ontology Matching (OM 2008) co-located with the 7th International Semantic Web Conference (ISWC 2008)*, Karlsruhe, Germany, 2008.

A. Ferrara, S. Montanelli, G. Varese, and S. Castano. Ontology Knowledge Authoring by Natural Language Empowerment. In *Proc. of the 1st International Workshop on Modelling and Visualization of XML and Semantic Web Data (MoViX 2009) co-located with the 20th International Conference on Database and Expert Systems Applications (DEXA 2009)*, Linz, Austria, 2009.

A. Ferrara, S. Montanelli, G. Varese, and S. Castano. Natural Language Ontology Authoring in iCoord. *International Journal of Information Technology and Database Systems (IJITDAS)*, 1(1), 2010.

F. Frasincar, R. Telea, and G. Houben. Adapting Graph Visualization Techniques for the Visualization of RDF Data. In V. Geroimenko and C. Chen, editors, *Visualizing the Semantic Web*. Springer-Verlag, 2006.

E. Gabrilovich, S. Dumais, and E. Horvitz. Newsjunkie: Providing Personalized Newsfeeds via Analysis of Information Novelty. In *Proc. of the 13th International Conference on World Wide Web (WWW 2004)*, New York, NY, USA, 2004.

Y. Gil and D. Artz. Towards Content Trust of Web Resources. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(4), 2007.

K.I. Goh, E. Oh, B. Kahng, and D. Kim. Betweenness Centrality Correlation in Social Networks. *Physical Review E*, 67(1), 2003.

O. Görlitz, S. Sizov, and S. Staab. PINTS: Peer-to-Peer Infrastructure for Tagging Systems. In *Proc. of the 7th International Workshop on Peer-to-Peer Systems (IPTPS 2008)*, Tampa Bay, USA, 2008.

L. Gu, R.A. Baxter, D. Vickers, and C. Rainsford. Record Linkage: Current Practice and Future Directions. Technical report, CSIRO Mathematical and Information Sciences, 2003.

S. Guha, N. Koudas, A. Marathe, and D. Srivastava. Merging the Results of Approximate Match Operations. In *Proc. 30th International Conference on Very Large Databases (VLDB 2004)*, Toronto, Canada, 2004.

A. Gullí. The Anatomy of a News Search Engine. In *Proc. of the 14th International Conference on World Wide Web (WWW 2005)*, Chiba, Japan, 2005.

R. Hahn, C. Bizer, C. Sahnwaldt, C. Herta, S. Robinson, M. Brgle, H. Dwiger, and U. Scheel. Faceted Wikipedia Search. In *Proc. of the 13th International Conference on Business Information Systems (BIS 2010)*, 2010.

M. Hausenblas. Exploiting Linked Data For Building Web Applications. *IEEE Internet Computing*, 13(4), 2009.

M.A. Hernández, S. Falconer, M. Storey, S. Carini, and I. Sim. Synchronized Tag Clouds for Exploring Semi-Structured Clinical Trial Data. In *Proc. of the Conference of the Center for Advanced Studies on Collaborative Research (CASCON 2008)*, Richmond Hill, Ontario, Canada, 2008.

M.A. Hernández and S.J. Stolfo. The Merge/Purge Problem for Large Databases. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, San Jose, California, USA, 1995.

M.A. Hernández and S.J. Stolfo. Real-World Data is Dirty: Data Cleansing and the Merge/Purge Problem. *Data Mining and Knowledge Discovery*, 2(1), 1998.

P. Heymann and H. Garcia-Molina. Collaborative Creation of Communal Hierarchical Taxonomies in Social Tagging Systems. Technical report, Computer Science Department, Standford University, 2006.

P. Heymann, G. Koutrika, and H. Garcia-Molina. Can Social Bookmarking Improve Web Search? In *Proc. of the International Conference on Web Search and Web Data Mining (WSDM 2008)*, New York, NY, USA, 2008.

C. Hirsch, J. Hosking, and J. Grundy. Interactive Visualization Tools for Exploring the Semantic Graph of Large Knowledge Spaces. In *Proc. of the Workshop on Visual Interfaces to the Social and the Semantic Web (VISSW 2009)*, 2009.

O. Hoeber and X. Yang. User-oriented Evaluation Methods for Interactive Web Search Interfaces. In *Proc. of the IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, Washington, DC, USA, 2007.

L. Hollink, M. van Assem, S. Wang, A. Isaac, and G. Schreiber. Two Variations on Ontology Alignment Evaluation: Methodological Issues. In *Proc. of the 5th European Semantic Web Conference (ESWC 2008)*, Tenerife, Spain, 2008.

A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. FolkRank: A Ranking Algorithm for Folksonomies. In *Proc. of the Workshop on Information Retrieval of the Special Interest Group on Information Retrieval (FGIR 2006)*, 2006a.

A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. The Semantic Web: Research and Applications. In Y. Sure and J. Domingue, editors, *Information Retrieval in*

Folksonomies: Search and Ranking, volume 4011 of *Lecture Notes in Computer Science*. Springer-Verlag, 2006b.

A. Isaac, H. Matthezing, L. van der Meij, S. Schlobach, S. Wang, and C. Zinn. Putting Ontology Alignment in Context: Usage Scenarios, Deployment and Evaluation in a Library Case. In *Proc. of the 5th European Semantic Web Conference (ESWC 2008)*, Tenerife, Spain, 2008.

A. Isaac, L. Van der Meij, S. Schlobach, and S. Wang. An Empirical Study of Instance-Based Ontology Matching. In *Proc. of the 6th International Semantic Web Conference (ISWC 2007)*, Busan, Korea, 2007.

J.M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46(5), 1999.

G. Koutrika, B. Bercovitz, R. Ikeda, F. Kaliszan, H. Liou, Z. Zadeh, and H. Garcia-Molina. Social Systems: Can We Do More Than Just Poke Friends? In *Proc. of the 4th Biennial Conference on Innovative Data Systems Research (CIDR 2009)*, Asilomar, CA, USA, 2009a.

G. Koutrika, Z. Zadeh, and H. Garcia-Molina. Data Clouds: Summarizing Keyword Search Results over Structured Data. In *Proc. of the 12th International Conference on Extending Database Technology (EDBT 2009)*, Saint Petersburg, Russia, 2009b.

B. Kuo, T. Hentrich, B. Good, and M. Wilkinson. Tag Clouds for Summarizing Web Search Results. In *Proc. of the 16th International Conference on World Wide Web (WWW 2007)*, Banff, Alberta, Canada, 2007.

A. Langegger, W. Wöß, and M. Blöchl. A Semantic Web Middleware for Virtual Data Integration on the Web. In *Proc. of the 5th European Semantic Web Conference (ESWC 2008)*, Tenerife, Spain, 2008.

D. Laniado, D. Eynard, and M. Colombetti. Using WordNet to Turn a Folksonomy into a Hierarchy of Concepts. In *Proc. of the 4th Workshop on Semantic Web Applications and Perspectives (SWAP 2007)*, Bari, Italy, 2007.

A. Leclercq. The Perceptual Evaluation of Information Systems using the Construct of User Satisfaction: Case Study of a Large French Group. *SIGMIS Database*, 38 (2), 2007.

V.I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10, 1966.

X. Li, J. Yan, Z. Deng, L. Ji, W. Fan, B. Zhang, and Z. Chen. A Novel Clustering-Based RSS Aggregator. In *Proc. of the 16th International Conference on World Wide Web (WWW 2007)*, Banff, Alberta, Canada, 2007.

H. Lin, J. Davis, and Y. Zhou. An Integrated Approach to Extracting Ontological Structures from Folksonomies. In *Proc. of the 6th Annual European Semantic Web Conference (ESWC 2009)*, 2009.

J. Madhavan, S. Jeffery, S. Cohen, X. Dong, D. Ko, C. Yu, and A. Halevy. Web-Scale Data Integration: You Can Only Afford to Pay as You Go. In *Proc. of the 3rd Biennial Conference on Innovative Data Systems Research (CIDR 2007)*, Asilomar, CA, USA, 2007.

C.D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

G. Marchionini. Exploratory Search: from Finding to Understanding. *Communications of the ACM*, 49(4), 2006.

B. Markines, C. Cattuto, F. Menczer, D. Benz, A. Hotho, and G. Stumme. Evaluating Similarity Measures for Emergent Semantics of Social Tagging. In *Proc. of the 18th International Conference on World Wide Web (WWW 2009)*, 2009.

P. Mika. Ontologies Are Us: A Unified Model of Social Networks and Semantics. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(1), 2007.

G.A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38, 1995.

R. Mirizzi, A. Ragone, T. Di Noia, and E. Di Sciascio. Semantic Wonder Cloud: Exploratory Search in DBpedia. In *Proc. of the 10th International Conference on Current Trends in Eeb Engineering (ICWE 2010)*, Berlin, Heidelberg, 2010.

S. Montanelli, S. Castano, A. Ferrara, and G. Varese. Managing Collective Intelligence in Semantic Communities of Interest. *International Journal of Organizational and*

Collective Intelligence (IJOCI), Special Issue on Collectively Intelligent Information and Knowledge Management, 1(4), 2010.

P. Mutton and J. Golbeck. Visualization of Semantic Metadata and Ontologies. In *Proc. of the 7th International Conference on Information Visualization*, Washington, DC, USA, 2003.

G. Navarro. A Guided Tour to Approximate String Matching. *ACM Computing Surveys*, 33(1), 2001.

H.B. Newcombe. *Handbook of Record Linkage*. Oxford University Press, Inc., 1988.

M.E.J. Newman. A Measure of Betweenness Centrality based on Random Walks. *Social Networks*, 27(1), 2005.

A. Nikolov, V.S. Uren, E. Motta, and A.N. De Roeck. Handling Instance Coreferencing in the KnoFuss Architecture. In *Proc. of the 1st ESWC International Workshop on Identity and Reference on the Semantic Web (IRSW 2008)*, Tenerife, Spain, 2008.

T. Opsahl, F. Agneessens, and J. Skvoretz. Node Centrality in Weighted Networks: Generalizing Degree and Shortest Paths. *Social Networks*, 32(1), 2010.

E. Oren, R. Delbru, and S. Decker. Extending Faceted Navigation for RDF Data. In *Proc. of the 5th International Semantic Web Conference (ISWC 2006)*, 2006.

H. Pasula, B. Marthi, B. Milch, S.J. Russell, and I. Shpitser. Identity Uncertainty and Citation Matching. In *Proc. of the Conference on Advances in Neural Information Processing Systems (NIPS 2002)*, Vancouver, BC, Canada, 2002.

J. Porter. *Designing for the Social Web*. New Riders Press, 2008.

D. Radev, J. Otterbacher, A. Winkel, and S. Blair-Goldensohn. NewsInEssence: Summarizing Online News Topics. *Communications of the ACM*, 48(10), 2005.

E. Rahm and P.A. Bernstein. A Survey of Approaches to Automatic Schema Matching. *The VLDB Journal*, 10(4), 2001.

G. Salton. *Automatic Text Processing - The Transformation, Analysis and Retrieval of Information by Computer*. Addison-Wesley, 1989.

G. Salton and C. Buckley. Readings in Information Retrieval. In *Term-Weighting Approaches in Automatic Text Retrieval*. Morgan Kaufmann Publishers Inc., 1997.

S. Sarawagi and A. Bhamidipaty. Interactive Deduplication Using Active Learning. In *Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2002)*, Edmonton, Alberta, Canada, 2002.

P. Schmitz. Inducing Ontology from Flickr Tags. In *Proc. of the Collaborative Web Tagging Workshop co-located with the 15th International Conference on World Wide Web (WWW 2006)*, Edinburgh, Scotland, 2006.

P. Shvaiko and J. Euzenat. A Survey of Schema-based Matching Approaches. *Journal on Data Semantics (JoDS)*, 4, 2005.

B. Sigurbjörnsson and R. Van Zwol. Flickr Tag Recommendation Based on Collective Knowledge. In *Proc. of the 17th International Conference on World Wide Web (WWW 2008)*, New York, NY, USA, 2008.

P. Singla and P. Domingos. Multi-Relational Record Linkage. In *Proc. of the 3rd KDD Workshop on Multi-Relational Data Mining*, Seattle, WA, USA, 2004.

S. Sorrentino, S. Bergamaschi, M. Gawinecki, and L. Po. Schema Normalization for Improving Schema Matching. In *Proc. of the 28th International Conference on Conceptual Modeling (ER 2009)*, Gramado, Brazil, 2009.

L. Specia and E. Motta. Integrating Folksonomies with the Semantic Web. In *Proc. of the 4th European Semantic Web Conference (ESWC 2007)*, 2007.

H. Stuckenschmidt, F. Van Harmelen, A. De Waard, T. Scerri, R. Bhogal, J. Van Buel, I. Crowlesmith, C. Fluit, A. Kampman, J. Broekstra, and E. Van Mulligen. Exploring Large Document Repositories with RDF Technology: The DOPE Project. *IEEE Intelligent Systems*, 19(3), 2004.

Y. Tian, R. Hankins, and J. Patel. Efficient Aggregation for Graph Summarization. In *Proc. of the 6th ACM SIGMOD International Conference on Management of Data (SIGMOD 2008)*, Vancouver, BC, Canada, 2008.

G. Tummarello, R. Cyganiak, M. Catasta, S. Danielczyk, R. Delbru, and S. Decker. Sig.ma: Live Views on the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(4), 2010.

G. Tummarello, R. Delbru, and E. Oren. Sindice.com: Weaving the Open Linked Data. In *Proc. of the 6th International Semantic Web Conference (ISWC 2007)*, Busan, South Korea, 2007.

G. Tummarello and C. Morbidoni. Collaboratively Building Structured Knowledge with DBin: From del.icio.us Tags to an "RDFS Folksonomy". In *Proc. of the Workshop on Social and Collaborative Construction of Structured Knowledge (CKC 2007) at WWW 2007*, Banff, Alberta, Canada, 2007.

G. Varese. Semantic Data Clouding over the Webs. In *Proc. of the 18th CAiSE Doctoral Consortium 2011 co-located with the 23rd International Conference on Advanced Information Systems Engineering (CAiSE 2011)*, London, United Kingdom, 2011.

G. Varese and S. Castano. Building Collective Intelligence through Folksonomy Coordination. In N. Bessis and F. Xhafa, editors, *Next Generation Data Technologies for Collective Computational Intelligence*, volume 352 of *Studies in Computational Intelligence*. Springer-Verlag, 2011.

V.S. Verykios, A.K. Elmagarmid, and E.N. Houstis. Automating the Approximate Record-Matching Process. *Information Sciences - Informatics and Computer Science: An International Journal*, 126(1), 2000.

C. Wang, J. Lu, and G. Zhang. Integration of Ontology Data through Learning Instance Matching. In *Proc. of the IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006)*, Washington, DC, USA, 2006.

Y.R. Wang and S.E. Madnick. The Inter-Database Instance Identification Problem in Integrating Autonomous Systems. In *Proc. of the 5th International Conference on Data Engineering (ICDE 1989)*, Washington, DC, USA, 1989.

T. Washio and H. Motoda. State of the Art of Graph-based Data Mining. *ACM SIGKDD Explorations Newsletter*, 5(1), 2003.

R. Wetzker, C. Zimmermann, and C. Bauckhage. Detecting Trends in Social Bookmarking Systems: a del.icio.us Endeavor. *International Journal of Data Warehousing and Mining*, 6(1), 2010.

W.E. Winkler. The State of Record Linkage and Current Research Problems. Technical report, Statistical Research Division, U.S. Bureau of the Census, 1999.

W.E. Winkler. Frequency-Based Matching in Fellegi-Sunter Model of Record Linkage. Technical report, US Bureau of the Census, 2000.

S. Yan, D. Lee, M.Y. Kan, and L.C. Giles. Adaptive Sorted Neighborhood Methods for Efficient Record Linkage. In *Proc. of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2007)*, Vancouver, BC, Canada, 2007.

K. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2003.

O.B. Yitzhak, N. Golbandi, N. Har'el, R. Lempel, A. Neumann, S.O. Koifman, D. Sheinwald, E. Shekita, B. Sznajder, and S. Yogev. Beyond Basic Faceted Search. In *Proc. of the International Conference on Web Search and Web Data Mining (WSDM 2008)*, New York, NY, USA, 2008.

R. Zhou and E. A. Hansen. Domain-Independent Structured Duplicate Detection. In *Proc. of the 21st National Conference on Artificial Intelligence (AAAI 2006)*, Boston, Massachusetts, USA, 2006.