Similarity Assessment of Data in Semantic Web

by

Parisa Dehleh Hossein Zadeh

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Software Engineering and Intelligent Systems

Department of Electrical and Computer Engineering

University of Alberta

Abstract

The web is a constantly growing repository of information. Enormous amount of available information on the web creates a demand for automatic ways of processing and analyzing data. One of the most common activities performed by these processes is comparison of data – it is done to find something new or confirm things we already know. In each case there is a need for determining similarity between different objects and pieces of information. The process of determining similarity seems to be relatively easy when it is done for a numerical data, but it is not so in the case of a symbolic data. In order to make the data stored on the Internet more accessible, a new model of data representation has been introduced – Resource Description Framework. Linked data provides an open platform for representing and storing structured data as well as ontology. This aspect of data representation has been fully utilized for providing fundamentals for the new forms of Internet, Linked Data and Semantic Web.

In this thesis, we investigate the problem of determining semantic similarity between entities in which not just lexical and syntactical information of entities are used, but the whole existing knowledge structure including the instantiated ontology is exploited. The idea is based on the fact that entities are interconnected and their semantics is defined via their connections to other entities as well as the metadata expressed as ontology. We propose feature-based methods for similarity assessment of concepts represented in ontology as well as in a less constrained Resource Description Framework. Membership functions are used to capture the importance of connections between entities at different hierarchy levels in ontology. We

leverage importance weighted quantifier guided operator to aggregate the similarity values related to different groups of properties. In another proposed approach, we use concepts of possibility theory to determine lower and upper bounds of similarity intervals. In addition, we address contextual similarity assessment when only specific context is taken into consideration. The idea of ranking entities' features according to their importance in describing an entity is introduced. We propose an approach that calculates similarly measures for these categories of features and then aggregates them using fuzzy-expressed weights that represents rankings of these categories. The promising results of our developed similarity method have encouraged us to extend it to a more comprehensive approach. As a result, we propose a technique for automatic identification of the importance of features and ranking them accordingly. Finally, we tackle the problem of application of heterogeneous feature types for defining entities. A method is described utilizing fuzzy set theory and linguistic aggregation to compare features of different types. We deploy this technique in a practical pharmaceutical application, where the proposed similarity assessment is shown to be capable of finding relevant entities – drugs in this case, in spite of heterogeneous features used to define them.

Preface

All of the research conducted for this thesis has been published in forms of journal articles and conference proceedings. All research results are the contribution of myself and my PhD supervisor, Professor Marek Z. Reformat at University of Alberta. The corresponding publication for each chapter is as follows:

Chapter 4 of this thesis has been published as P. D. Hossein Zadeh and M. Z. Reformat. (2013) Assessment of semantic similarity of concepts defined in ontology. International Journal of Information Sciences, Elsevier. volume 250. pp: 21-39. (published)

Chapter 5 of this thesis has been published as P. D. Hossein Zadeh and M. Z. Reformat. (2012) Feature-based Similarity Assessment in Ontology using Fuzzy Set Theory. IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). pp: 1-7, 2012. (published)

Chapter 6 of this thesis has been published as P. D. Hossein Zadeh and M. Z. Reformat. (2013) Context-aware Similarity Assessment within Semantic Space Formed in Linked Data. Journal of Ambient Intelligence and Humanized Computing. volume 4, issue 4. pp. 515-532. (published)

Chapter 7 of this thesis has been published as P. D. Hossein Zadeh and M. Z. Reformat. (2012) Fuzzy Semantic Similarity in Linked Data using the OWA Operator. 2012 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS). pp: 1-6. (published)

Chapter 8 of this thesis has been published as: 1- P. D. Hossein Zadeh and M. Z. Reformat. (2013) Fuzzy Semantic Similarity in Linked Data using Wikipedia Infobox. IEEE IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS) Joint. pp: 395-400. (published)

2- P. D. Hossein Zadeh and M. Z. Reformat (2015) The Web, Similarity, and Fuzziness. 50 Years of Fuzzy Logic and its Applications, Springer. volume 326, pp: 519-536. (published)

Chapter 9 of this thesis has been published as P. D. Hossein Zadeh, M. D. Hossein Zadeh, M. Reformat, (2015) Feature-driven Linguistic-based Entity Matching in Linked Data with Application in Pharmacy, Soft Computing Journal, Springer. pp: 1-16. (published). M. D. Hossein Zadeh is a clinical Pharmacist and assisted us during the methodology design and experimental studies with pharmacy technical help and valuable feedbacks.

This thesis is dedicated to my lovely parents, Shahnaz and Hamid

Acknowledgements

First, I would like to sincerely thank my PhD supervisor, Prof. Marek Z. Reformat for all his effort, encouragements, and passion. Thank you for believing in me and giving a wonderful experience of being your PhD student.

I would like to thank my parents, Shahnaz and Hamid for their support and endless love. I would like to thank my sister and brother-in-law, Mahsa and Peyman for being so caring and helpful to me.

I have learnt so much from all of you in my life. Thank you!

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Linked data - LD

Linked Open data - LOD

Semantic Web - SW

Resource Description Framework- RDF

Uniform Resource Identifier – URI

Hypertext Transfer Protocol - HTTP

Ordered Weighting Averaging – OWA

World Health Organization - WHO

Chapter 1

1 Introduction

1.1 Overview

The fast growing number of web pages and available data creates demand for better ways of finding data that is interesting and useful for a user. Design of robust and effective tools to manage and facilitate access to information stored on the web becomes an important undertaking. Semantic similarity between two resources is critical to be determined in processes and applications such as information extraction and retrieval, automatic annotation, web search engines, ontology matching, etc. As an example, identification of the data satisfying user's request is realized by matching the query keywords to pieces of information such that most of the web search engines utilize this approach and its variations. In information systems, semantic similarity plays an important role that is based on assessing similarities between semantic units of resources in order to identify the resources that are conceptually relevant but not identical. In this thesis, we address the topic of determining semantic similarity, which becomes profoundly important and useful in many applications.

As mentioned, the web is a vast repository of distributed data while it grows with an astonishing rate. Different varieties of data formats are utilized, such as pictures, videos, music, symbolic, and numerical data. At the same time, users constantly search the web and expect perfect answers to their needs. Dependency of users on the web becomes more and more

pronounced. However, the variety of information available and stored on the web becomes a potential problem of how to search for things that are described with numerical and symbolic values, and how to ensure multiple aspects of the user request are satisfied. More often users are looking for less specific things. This applies especially to data of numerical nature. In many cases users do not care about the exact values, thus they query in an imprecise way using approximate values [76]. They use linguistic terms, such as “most”, “minimum” or even “safest”, or “related to”.

Internet is perceived as a source of multiple types of information, a large shopping mall, a social forum and an entertainment hub. Users constantly browse and search the web in order to find things of interest. The keyword-based search becomes less and less efficient in the case of more refined searches where details of items become important. The introduction of Resource Description Framework (RDF) as a relation-based format for data representation allows us to propose a different way of performing a relevancy-based search. Every day, millions of users search and browse the web. Besides news and information, they also look for items of possible interest: books, movies, hotels, travel destinations, and many more. It is anticipated that these items possess specific or similar features (Tversky 1977). Additionally, not all of these features are equally important, some of them are significant with a high selective power, while some are entirely negligible. The improvement of the users’ searching activities depends on development of web applications that are able to support the users in finding relevant entities. So far, identification of data satisfying user’s request is realized by

matching the query keywords to pieces of information. Most of the web search engines utilize this approach and its variations.

A novel representation of information on the web, introduced by the concept of Semantic Web by Berners-Lee in 2001, changes the way how individual pieces of information are stored and accessed on Internet. The fundamental data format is called RDF and it relies on a simple concept of a triple: <subject-property-object>. In other words, a triple can be perceived as a relation existing between two entities: one of them – subject – is the main entity that is in relation embodied by a property with another item – object. It means that any piece of information can be represented as a set of triples where multiple items are linked to each other being subjects and/or objects in different triples. The Semantic Web as an enhancement to the current web presents meaning and structure of the contents. Data in Semantic Web is described using RDF in a triple format of subject-property-object. Uniform Resource Identifiers (URI) are used to uniquely identify each subject and predicate. The object is either another URI or a literal such as a number or a string.

In other words, Semantic Web [38] is a new paradigm that provides a novel format for information representation on the web. An ultimate contribution of Semantic Web [38] is utilization of ontology as the knowledge representation form. The term ontology is used in two different ways representing two different things. In its first usage – philosophical ontology – ontology is a description of reality in terms of classification of reality [68]. In its second usage – ontology and information systems – ontology deals with taxonomy of terms that describe a certain area of knowledge. In this context, the most popular definition says "ontology is a

formal, explicit specification of a shared conceptualization" [26]. Since ontologies do more than just control a vocabulary, they are treated as a new form of knowledge representation.

This aspect of ontology has been fully utilized for providing fundamentals for the new form of Internet, Semantic Web. The ultimate goal of the Semantic Web is to deliver a new Internet providing an environment that is more suitable for automatic data discovery and service providing. It has become obvious that in order to make this possible a new way of representing information and knowledge is needed. At the same time, it has become more and more evident that the current way of storing data on the web would not lead to solutions of the issues raised above. This representation should be such that each software agent is able to read and understand any information that exists on the Internet. Therefore, the definition of a method for evaluating the semantic similarity in this environment utilizing this underlying knowledge and meaning of data becomes essential. This problem is recognized and W3C[1] has proposed a different way of representing information on the web, namely RDF [51].

In this format two entities are "connected" via a relation that exists between them. For example, the RDF triple "*John livesIn Edmonton*" links *John* and *Edmonton* via the relation *livesIn*. This relation describes the fact that there is some kind of relationship – *livesIn* – between entities called *John* and *Edmonton*. There may be two other triples of "*John is-a person*" and "*Edmonton is-a city*" that provide more information about *John* and *Edmonton*. This simple example gives a glimpse of powerful characteristics of RDF in which all the information is highly interconnected. Additionally, RDF has been defined in a way that any piece

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

of information can be stored at any location on the web. These two features alone create enormous opportunities for different semantic-oriented analysis and utilization of data stored on the web.

In fact, RDF [51] is introduced as an underlying framework in order to use ontology in the web environment. Graphically, RDF triples can be presented in a so-called RDF graph [51]. Figure 1.1 illustrates three triples describing the movie “The matrix”. In the last few years, the application of RDF for data representation has become a very popular way of representing data on the web [66].

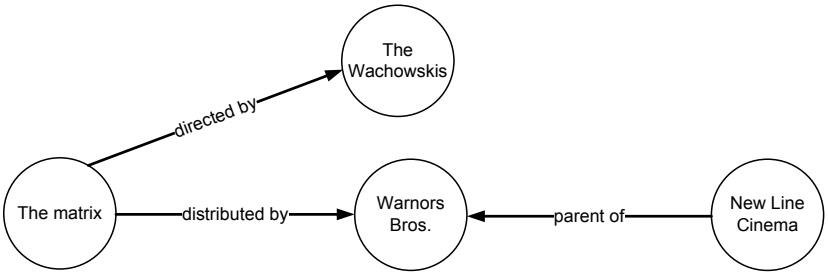


Figure 1.1 A simple example of RDF graph containing three triples

The growing amount of data stored in the RDF format on the web has led to create a form of the web called Linked Open Data (LD) [7]. The data in LD is highly interconnected. Over time, more attention has been paid to RDF data representation, and the term LD has been used to describe the network of data sources based on RDF triples for information representation [7]. The power of LD, in contrary to hypertext web, is that entities from different sources and locations are linked to other related entities on the web. This enables one to view the web as a single global data space [7]. In other words, hypertext web connects documents in a naïve way – links always point to documents. However, in the web of LD single information items are

connected – links point to other pieces of information stored at different physical locations. As a result, LD allows for better representation of structured data and even its underlying semantics.

In order to publish data on the web of LD some principles have to be followed [6]. One fundamental rule is the use of URIs to identify each piece of information [66]. URIs aim to universally define entities in the web of data such that users and machines can use the URIs to obtain information about the data. This means that every entity has a global identifier that a person or machine can use to look it up, refer to it, and find its description. Another rule of publishing data in LD is that the created URIs should be obtainable via Hypertext Transfer Protocol (HTTP) on the web. As stated, LD is expressed in RDF triples, where each one of the components is represented by an URI. This way of finding a specific piece of information in the web of data is facilitated with the help of interpretable URIs. For example, the entity “University of Alberta” can be referred to in different ways, such as “University of Alberta”, “UofA”, and “Ualbrta” by different data sources. However, assigning a unique URI in different datasets helps to avoid any confusion.

A collection of Semantic Web technologies and applications supports manifestation of LD in reality. These include protocols, strategies and tools for querying the RDF datasets (SPARQL, ViziQuer), transforming current application-specific formats of resources to the RDF format (Triplify, PhotoStuff, Virtuoso sponger, Csv2rdf), reasoning and discovering new relationships using RDF data in order to manage the information on the web (Jena, FaCT++), and extracting

RDF triples (3Store, Pubby). A special semantic query language SPARQL (SPARQL Protocol and RDF Query Language) is used to access data stored in RDF format.

Based on the Semantic Web vision, several knowledge bases have been created including DBpedia[2], Geonames[3], YAGO[4], FOAF[5], etc. This collection of interrelated datasets on the Web is also referred to as LD. DBpedia is a typical case of a large linked dataset, which transforms the contents of Wikipedia into RDF triples. Even though DBpedia is a large dataset and contains over one billion triples from Wikipedia data, it also provides RDF links to other datasets on the Web such as Geonames and Freebase[6].

1.2 Research motivation

An evaluation of relevancy between any two items on the web is associated with determining similarity between them. Therefore, similarity assessment between two items is an important and fundamental step in processes and applications related to information extraction and retrieval, web search, automatic annotation, database applications, etc. Although several achievements have been obtained for evaluation of similarity between entities on the web, there still exist many uncovered questions. For instant, many methods proposed for analysis and query of RDF data rely on taxonomies, which in many cases, taxonomies of different datasets are not comparable. Thus, ontology-based approaches encounter problems when concepts belong to different datasets in LD. Moreover, corpus-based

[2] http://dbpedia.org/About
[3] http://www.geonames.org/
[4] http://www.mpi-inf.mpg.de/yago-naga/yago/
[5] http://www.foaf-project.org/
[6] http://www.freebase.com/

methods have shown reasonable alignment with human judgment of similarity however these techniques focus on traditional information representation models such as web pages and documents, and are not suitable for capturing the semantics of data. This research is aimed at addressing some of these problems. We argue that similarity-based query and analysis of RDF data that take features of concepts[7] into consideration are best suited for the vast network of interconnected data, LD.

As mentioned before, LD is potentially beneficial to various applications such as web search engines, web browsers, information retrieval systems, and reasoning engines. Indeed, we can say that LD is a powerful infrastructure providing entities with semantic. As a result of the interconnected data, navigation and query using semantic-enabled browsers over the LD can be facilitated to a great extent. However, LD as an integration of the interlinking datasets poses challenges regarding processing and analysis of data [53]. One of them is semantic similarity discovering between pieces of information. Several approaches have been proposed for assessing the similarity between entities based on their lexical, taxonomic and information-based characteristics but a very little attention has been given to their underlying semantic. This research targets semantic similarity evaluation of the interconnected data in the level of LD and ontology.

In particular, connections represent reasonable amount of information about the entities in LD. In such representation of a single item, each triple is treated as its feature. Detailed analysis of these interconnections enables one to extract features related to every entity in the

[7] Throughout this chapter, two terms “concept” and “entity” are used interchangeably.

web of data. Application of RDF as data representation format allows us to propose novel approaches to evaluate items' relevancy. These facts motivate us to apply a feature-based comparison of items and to take advantage of its flexibility and adaptability in a process of evaluating relatedness between items. The above-mentioned principle sheds light on how to assess the degree of semantic similarity between two entities in LD. Moreover, the fact that human evaluates the similarity in the same fashion creates a natural implication of describing similarity as a feature-matching process.

1.3 Thesis contributions

The ultimate goal of this thesis is development of methodologies for evaluation of similarity of entities in ontology and LD environment. These methodologies are feature-driven and represent an improvement to the existing techniques for determining similarity of entities in ontology. Specifically, the proposed techniques allows for considering context, importance of features as well as matching heterogeneous features. Below, we briefly introduce and highlight the main contributions of this thesis, which can be categorized into four topics:

a) *Feature-based similarity with context-awareness:*

Based on the concept that semantic is formed through connections between resources on the web we proposed methods that treat the underlying infrastructure as a large semantic space containing multiple definitions. These approaches are based on the fact that an item is

represented as a set of triples while all of them are “tied” by the fact that a subject of these triples is the same, i.e., it is an item under consideration.

As mentioned before, ontology is a knowledge representation source that is widely used in Semantic Web to feed structured vocabulary in a relevant domain of topic. We proposed methods on semantic similarity between concepts defined in ontologies, which has a fundamental role in processing and analyzing data represented in ontologies. The measures that we propose in this report may also improve suggestions for merging and aligning ontologies. We presented a technique in Chapter 4 that focuses on more than just ontological information about concepts by considering the relations between concepts and their semantics. The method covers information in both the definition (abstract) level as well as instance level. The proposed methodology also allows for context-aware similarity assessment. One of the main contributions of this thesis is the evaluation and integration of context-awareness into measuring relevancy of two entities together. This thesis also reports on quantitative characterization and combination of similarity at different levels of abstraction in ontology using elements of fuzzy set theory, in Chapter 5.

Acknowledging the fact that not all datasets have ontologies associated with them, we broadened our scope to semantic similarity assessment in the web of linked data. As a result, we proposed multiple techniques (Chapters 6, 7, 8, 9) on semantic similarity assessment in linked data environment, where entities are submerged on linked data and their semantics is defined via their connections to other entities. Proposed approaches are based on

representation of items as sets of features. This means that evaluation of items' relevance is based on a feature-based comparison.

b) Categories of similarity

With regards to the web of linked data, we proposed a method to employ possibility theory to calculate the degree of similarity between two entities, Chapter 6. The underlying idea is to identify and categorize the connected entities of the resources under similarity study into entities that are certainly shared and possibly shared between the two resources, and uses elements of possibility theory to assess similarity between these entities. The proposed similarity method is extended in order to allow contexts to be considered, while definition and importance of context are discussed and evaluated. This method evaluates on every ordered pair of resources with necessity of similarity and possibility of similarity as lower-bound and upper-bound values. Usage of the ordered pairs creates an asymmetric way of calculating similarity.

c) Ranking of features

We argue that a more realistic relevancy determination can be been achieved via ranking of entities' features based on their importance, as presented in Chapters 7 and 8. To support our idea, we proposed novel approaches on levelling and ranking features based on their importance in describing the entity. In proposed methods, the calculated similarity measures for these categories of features are aggregated using fuzzy weights associated with the importance of these categories. Extending this research work, we proposed the idea of

automatically identifying the important features of an entity in linked data and ranking them according to their influence on the similarity measure. To our knowledge, there has been no report in literature on measurement of relevancy with such a ranked structures. The proposed method can be generalized and applied to any relevant research work in the literature.

d) Heterogeneous feature types

Finally, Chapter 9 of the present report includes a method for matching entities that are defined by features that are expressed in different formats such as numerical and symbolic. In this method and throughout this report, elements of fuzzy set theory and linguistic aggregation are applied to combine and compare entities in order to determine their similarity and satisfaction levels to the reference requirements. More specifically, the relevancy score for each type of feature is determined and mapped into a fuzzy universe. The next important step, which is the aggregation of individual matching scores is performed by adapting a 2-tuple linguistic representation model [30]. The application of 2-tuple enables us to deal with several different linguistic-based features. Also, we utilized a linguistic aggregation mechanism representing a special case of multi-criteria decision-making processes. We evaluated and deployed this technique in a practical application of Pharmacy, where features are in different formats and distributed over multiple datasets. To the best our knowledge, nothing has been reported on the heterogeneous feature-based similarity evaluation and usually homogenous feature types is assumed in the existing similarity metrics.

It worth noting that the computational complexity of the above proposed methods does not exceed O(n*m), where n and m are number of features of the two entities under study. This is mainly due to intrinsic complexity accompanied with feature-based techniques.

1.4 Outline and organization

The organization of this thesis can be summarized as follows: In Chapter 2, a detailed technical description of the similarity definition, linked data, RDF triples, ontology, along with similarity in each of them, and an introduction to linguistic aggregation technique is provided. These information and techniques are utilized in semantic similarity methods presented throughout Chapters 4 – 9. Chapter 3 includes a comprehensive literature review of the methodologies in assessing similarity of concepts in the current web, linked data as well as ontology. It contains numbers of approaches addressing the problem of similarity computation. In Chapter 3, we have categorized and discussed them in details based on the deployed methodology. Chapter 4 and 5 focus on similarity of entities in ontologies while Chapters 6 – 9 explore around semantic similarity in linked data.

The presented method in Chapter 4 not only takes into calculation the structured ontological information but it also explores the connections between concepts and their semantics. The method includes abstract level information as well as instance level. It also allows for context-aware similarity assessment.

Chapter 5 reports on quantitative characterization and combination of similarity at different levels of abstraction in ontology using elements of fuzzy set theory. Context-based similarity is discussed and comparison studies are presented.

In Chapter 6, possibility theory is used to calculate the degree of similarity between two entities in the web of linked data. The proposed similarity measure is further developed to include context, which also experimentally evaluates the importance of context.

The idea of categorization and ranking of features can be seen in Chapter 7 and 8. Our approach in Chapter 7 is based on the idea that features of an entity can be categorized and ranked based on their importance in describing the entity. The calculated similarity measures for each category of features are aggregated with the importance of that category and its associated fuzzy weights. In other words, aggregation of the calculated similarity measures and their fuzzy weights is performed. In Chapter 8, we propose an improvement to the presented method in Chapter 7, by automatically identifying the important properties of an entity and ranking them according to their influence on the similarity measure.

Chapter 9 reports on a developed methodology for finding relevancy in entities that are defined by features that are expressed in different formats such as numerical and symbolic. Concepts of fuzziness and linguistic aggregation are applied to combine and compare entities in order to determine their similarity levels to the referenced format requirements.

The conclusion of present dissertation and suggested future research work are presented in chapter 10.

Chapter 2

2 Background

Similarity is essential for finding relevant things. This can be further explored by a need to dig a bit "deeper" and look not only on items as whole units but also at individual features of these items. Potentially, this can lead to a more refined similarity estimation process and better results. The introduction of RDF provides an opportunity to "see" items as sets of features and build simple procedures for evaluating similarity of items.

The goal of the Semantic Web as an enhancement to the current web is to provide a meaning and structure to the web content. The Semantic Web's road map points to ontology as a way of accomplishing this. Ontology defines a structural organization and relations between concepts[8], properties and instances. It also adds semantic richness and reasoning capabilities. Any type of information expressed with a means of ontologies can be semantically analyzed and processed leading to more comprehensive results.

2.1 Definition of similarity

A variety of approaches for similarity assessment have been proposed in the literature while some leverage lexicographic, syntactic, structural information and representation of information about entities to measure the similarity. The most popular techniques are based on entities' feature matching [50, 72] as well as combination approaches [13, 69]. Many

[8] Throughout this chapter, two terms "concept" and "entity" are used interchangeably.

approaches depend on representation of information in the form of ontology, while a few methods investigate the problem of similarity assessment in LD.

The original definition of identity comes from the Leibniz's law of identity of indiscernible [39], which states that the two entities i and j are identical if they share common properties P_i and P_j:

$$\forall i \forall j [\forall P(P_i = P_j) \rightarrow i = j] \tag{2.1}$$

P_i and P_j refer to sets of properties for entities i and j, respectively. It can be inferred that unique features of each entity contribute to the dissimilarity measure between the two entities.

Another important aspect of such understood (dis)similarity is related to its symmetry. If number of features of an entity is different (or weighted differently) than another entity then (dis)similarity is not symmetric. This complies with the work conducted by Tversky [72]. We also believe that similarity can be determined when only some specific features are considered while others are meant to be ignored. Thus, an appropriate selection of features allows for determining similarity in a context defined by these selected features.

2.2 Linked open data

2.2.1 Linked data and RDF triple definition

Linked data (LD) resembles a decentralized partial mesh network in which entities from different resources are connected to other related entities directly or indirectly. In fact, all

pieces of information in LD are expressed using triples. The generic format of a RDF triple is: subject, property, and object. For example, the statement:

"The Matrix (movie) is distributed by Warner Bros."

can be expressed as the following triple:

The Matrix**(subject)-**distributed by**(property)-**Warner Bros.**(object)**

Graphically, the above RDF triple is constituted in a so called RDF graph, see Figure 2.1.

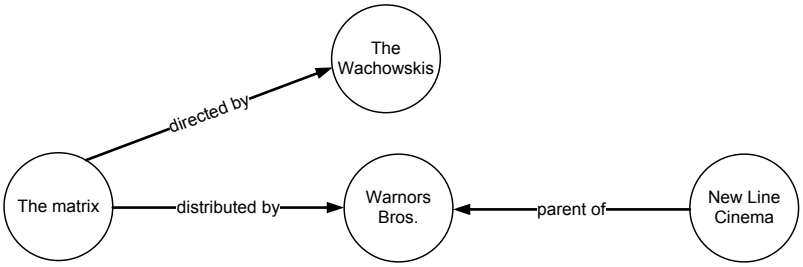


Figure 2.1 A simple RDF graph containing three triples

Dereferencing the URI associated with every entity enables a user/machine to find all information related to that entity, which includes its associative RDF fragments in the web of data.

There exist several important data collections that have published their contents in the format of LD, such as DBPedia[9], Geonames[10], Freebase[11], New York Times[12], BBC programmes[13], and FOAF[14].

[9] http://dbpedia.org/About

[10] http://www.geonames.org/

Figure 2.2 is generated using Gephi[15] and depicts a snapshot of DBPedia dataset containing RDF triples of four different movies. Vertices represent resources (subjects and objects), and properties are shown by edges between the resources. One of the most intriguing observations regarding LD is its contribution to semantic definition of entities. A set of relations between an entity and other resources can be conceived as resource's features defining its semantics.

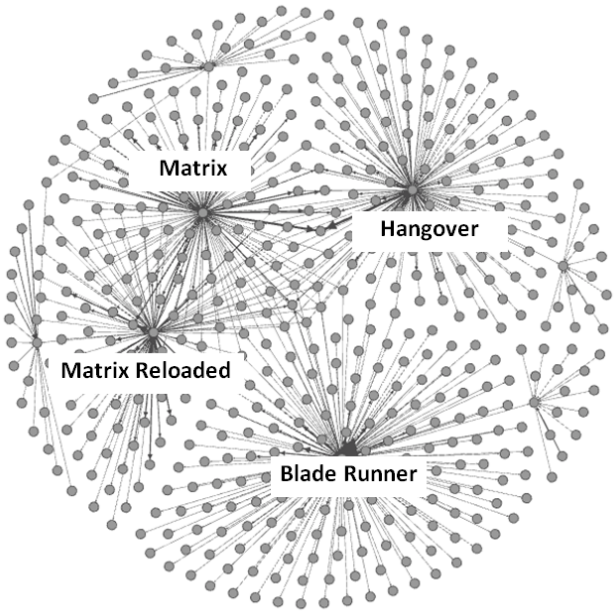


Figure 2.2 A snapshot of dbpedia.org dataset representing four movies

The Semantic Web concept introduces RDF as a way of representing information including ontologies and their instances. The fundamental idea is to represent each piece of data as a

[11] http://www.freebase.com/
[12] http://data.nytimes.com/
[13] http://www.bbc.co.uk/programmes
[14] http://www.foaf-project.org/
[15] http://gephi.org/

triple: <subject-property-object>, where the subject is an entity being described, the object is an entity describing the subject, and the property is a “connection” between the subject and object. For example, *Godfather is book* is a triple with *Godfather* as its subject, *is* its property, and *book* its object. In general, a subject of one triple can be an object of another triple, and vice versa. The growing presence of RDF as a data representation format on the web brings opportunity to develop new ways of how data is processed, and what type of information is generated from data.

A single RDF-triple <subject-property-object> can be perceived as a feature of an entity identified by the subject. In other words, each single triple is a feature of its subject. Multiple triples with the same subject constitute a definition of a given entity. A simple illustration of this is shown in Figure 2.3.a. It is a definition of Godfather. If we visualize it, definition of the entity resembles a star with the defined objects as its core. We can refer to it as an RDF-star.

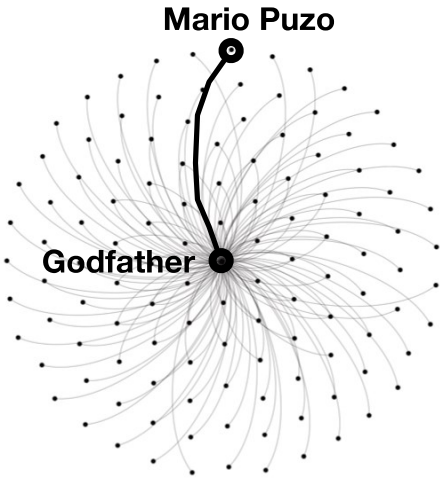
Quite often a subject and object of one triple can be involved in multiple other triples, i.e., they can be objects or subjects of other triples. In such a case, multiple definitions – RDF-stars – can share features, or some of the features can be centres of another RDF-stars. Such interconnected triples constitute a network of interleaving definition of entities, Figure 2.3.b.

In general, Uniform Resource Identifiers (URI) are used to uniquely identify subjects and properties. Objects, on the other hand, are either URIs or literals such as numbers or strings.

Due to the fact that everything is interconnected, we can state that numerous entities share common features. In such a case, comparison of entities is equivalent to comparison of

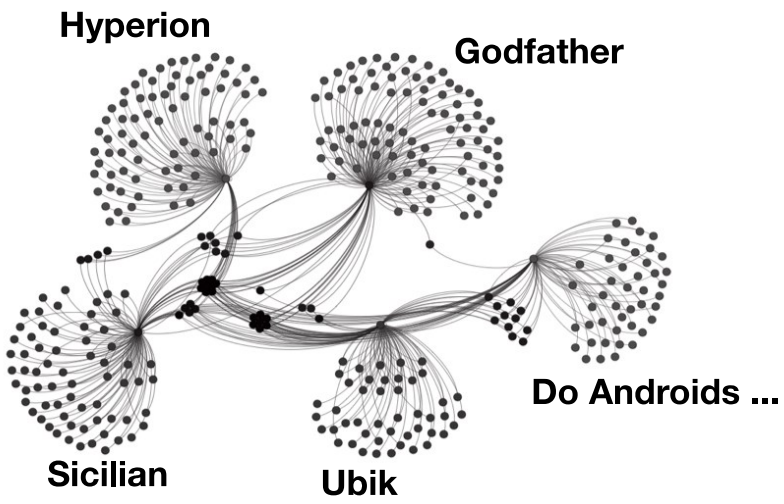
RDF-stars. This idea is a pivotal aspect of the proposed approach for determining relevance of items.

Based on the Semantic Web vision, several knowledge bases have been created including DBpedia[16], Geonames[17], YAGO[18], and FOAF[19]. The collection of interrelated datasets on the Web is referred to as Linked Open Data (LOD) (Bizer and Berners-Lee 2009). DBpedia is a large linked dataset, which contains Wikipedia data translated into RDF triples. Even though DBpedia is a large dataset that has over one billion triples from Wikipedia, it also provides RDF links to other datasets on the Web such as Geonames and Freebase[20]. With a growing number of RDF triples on the web – more than 62 billions[21] triples– processing data in RDF format is gaining special attention. There are multiple work focusing on RDF data storage and querying strategies using a specialized query language SPARQL.



[16] http://dbpedia.org/About
[17] http://www.geonames.org/
[18] http://www.mpi-inf.mpg.de/yago-naga/yago/
[19] http://www.foaf-project.org/
[20] http://www.freebase.com/
[21] http://stats.lod2.eu

(a)



(b)

Figure 2.3 (a) RDF-stars: a definition of Godfather with one of its features enhanced, (b) interconnected RDF-stars representing: Godfather, Hyperion, The Sicilian, Ubik and Do Androids Dream of Electric Sheep.

2.2.2 Linked data and similarity

Information in LD is represented in RDF format, i.e., triples: subject, property, and object, where each of these items is identified by an URI (Uniform Resource Identifier). The nature of LD is that entities from different datasets are linked together. One important task in LD is the assessment of semantic similarity between two entities. This is a critical step in many processes and applications such as automatic annotation, web search engines, personalization on the web, recommender systems, ontology matching, and information extraction and retrieval. In answering a user-defined query, the query keywords are matched to the information on the web. In literature, several approaches for similarity assessment have been proposed where

some are based on lexicographic, syntactic and structural information. Feature-based matching of entities is introduced by [72] that evaluates the similarity by comparing features of entities. Features associated with entities can be translated as RDF triples, where subjects and properties are entities and features accordingly. We believe that in calculating the similarity some important features should be considered while others can be ignored. In this process, identifying the key features of an entity is crucial [35]. We also believe that it is important to assess the similarity of each feature according to its context.

Interesting approaches for evaluating similarity in LD have been proposed in [5, 12, 64]. Soft computing and reasoning on web contents for knowledge discovery is presented in [43], while the semantic mapping of concepts in ontology is the topic of the work presented in [19]. For extensive review of current semantic similarity techniques see Chapter 3 or [11].

RDF data representation introduced by the Semantic Web community leads to an important observation that is a principal idea of the proposed approach: similarity between pieces of information can be determined by analysis of connections between these pieces and other information.

2.3 Ontology

An ontology deals with a taxonomy of terms that describe a certain area of knowledge. Since ontologies do more than just control a vocabulary, they are treated as a new form of knowledge representation. This aspect of ontology has been fully utilized for providing fundamentals for a new form of Internet – the Semantic Web. It can be seen that the definition

of ontology provides a significant set of interconnections that semantically define a concept. The ontology provides semantic interconnection of a given concept via connecting it with all concepts that are relevant. Indeed, these connections constitute features of the defined concept. The most important aspect of ontology used for the Semantic Web applications is to identify two ontology layers: the ontology definition layer and the ontology instance layer. According to the latest terminology, the term "instance" has been replaced by "individual".

2.3.1 Ontology definition

The ontology definition layer represents a framework for establishing a structure of ontology and for describing each concept (node) in it. A structure of ontology is built based on is-a relation between nodes. This relation represents a subClassOf connection between a superclass node and a subclass node. In such a way, a hierarchy of concepts (nodes) is built. There are two main types of properties in ontology: datatype properties and object properties. Both of them provide a way of accurate and complete description of a concept (node).

2.3.2 Ontology individuals

Once the ontology definition is constructed, its individuals can be built. It means that the properties of the nodes are initialized; datatype properties are filled out with specific values, and object properties are linked to individuals that are instances of other nodes. An example of ontology instance is presented in Figure 2.4.

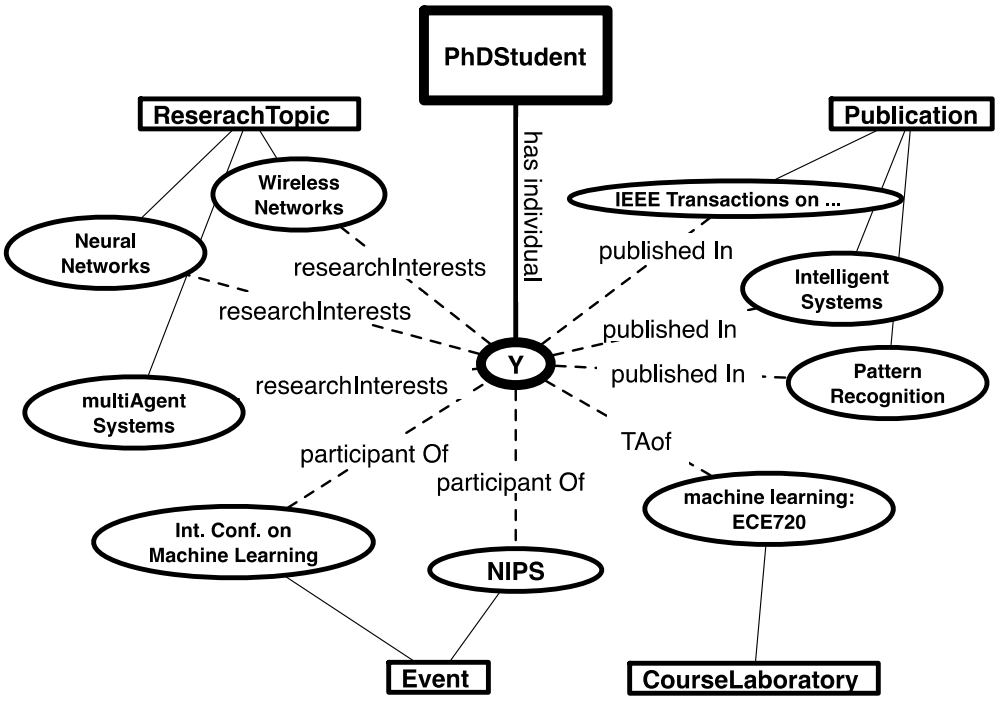


Figure 2.4 Ontology Instance: defining the PhD student *Y*

2.3.3 Ontology and similarity

Ontology consists of a finite set of concepts and the relationships between them in order to describe a certain area of knowledge. Information in ontology is structured in a hierarchical way created by domain experts. This new aspect of knowledge representation is utilized by Semantic Web with the goal of providing an environment that is suitable for machine/user data processing.

One of the important aspects of ontology used for Semantic Web applications is that knowledge in ontology is represented at different levels of details starting from an abstract definition level, and finishing at an instance level. The definition level contains general information about concepts, their features and relationships to other concepts. The instance

level describes information about individuals or instances of the concepts described in the definition level.

The underlying idea for calculating similarity is that more abstract concepts are located at higher locations in ontology. According to the principles in information theory it is known that the more abstract concepts have lower information contents [62]. In other words, interconnections between concepts located in different levels of abstraction can carry different similarity distances between the concepts. It should be emphasized that interconnections contain all types of relations between concepts in ontology.

Semantic distance according to the level of abstraction in ontology can be seen in many similarity assessment approaches, which significantly improves the matching results and make them closer to the human judgment. For example, in information-theoretic models, the information content of the least common super concept plays a key role in measuring the similarity [62]. Also, in structural models of similarity assessment [74] the use of hierarchical information of concepts is necessary for calculating the similarity measure.

2.4 Linguistic aggregation

An important step of any multi-criteria decision-making process is an aggregation of individual scores representing levels of satisfaction of each criterion. Such a process is equally important in the case of finding an entity that matches multiple requirements to the highest degree. In this process, we deal with numeric and symbolic values representing levels of

satisfaction; thus, we have adopted a 2-tuple linguistic representation model proposed in [30] in Chapter 9.

The linguistic model is based on representing linguistic information as 2-tuples. It means that satisfaction values of different criteria are expressed as pairs of: a fuzzy linguistic term and a numeric value in the range [-0.5, 0.5). The reason for adopting this approach for processing linguistically represented data is twofold. First, we deal with real-life problems where information can be better presented in an approximate and qualitative form rather than a fixed and quantitative way. Second, it reduces the information loss by means of representing the information and the results of computation, i.e., aggregation, in a continuous manner.

The application of 2-tuple fuzzy linguistic representation model implies that information is represented by 2-tuples (t, α), where *t* is a linguistic term defined in the universe of discourse U, and α is a numeric value in the interval [-0.5, 0.5). The linguistic terms $T = \{t_1, t_2, \ldots, t_n\}$ defined on U represent degrees of satisfaction of a specific criterion, e.g., *low*, *medium*, and *high*. The terms are defined such that they fully express semantics of the domain. The numeric value α represents a "deviation" from the value that is a numeric center of a linguistic term. In the research work in Chapter 9, we use triangular membership functions for the set of seven linguistic terms as follows (see Figure 2.5 and Figure 2.6):

$$T = \begin{Bmatrix} t_0 = extemely\ low(EL), t_1 = very low(VL), t_2 = low(L), t_3 = medium(M), t_4 = high(H), \\ t_5 = very high(VH), t_6 = extremely\ high(EH) \end{Bmatrix}$$

where each has a range as follows:

EL=(0,0,0.17)	VL=(0,0.17,0.33)
L=(0.17,0.33,0.5)	M=(0.33,0.5,0.67)
H=(0.5,0.67,0.83)	VH=(0.67,0.83,1)
EH=(0.83,1,1)	

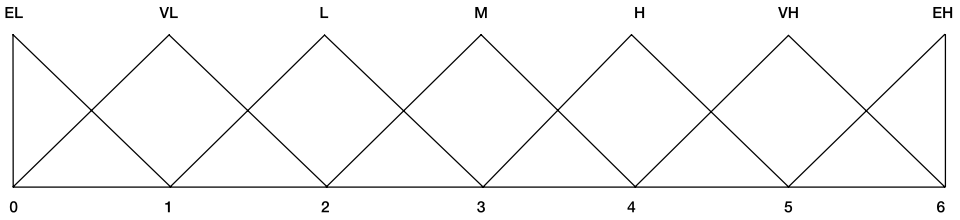


Figure 2.5 Linguistic terms t_0 to t_6 (EL - extremely low to EH – extremely high) defined in the universe of discourse *<0,6>* required for linguistic aggregation

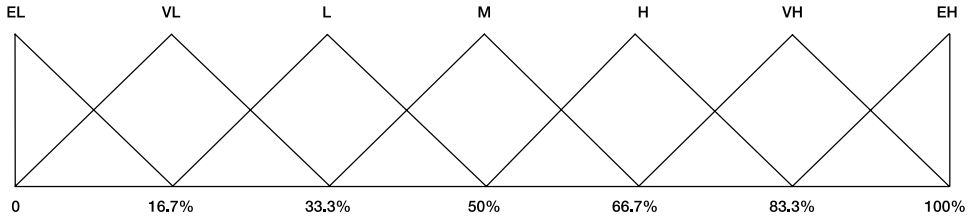


Figure 2.6 Linguistic terms defined in the universe of discourse <0,100%> for a case of numeric input.

The process of translating the result of aggregation into a 2-tuple is done in the following way: Let $\beta \in [0,6]$ represents the result of aggregation operation. The index *i* of a linguistic term, t_i, is determined as $\mathrm{i} = \mathrm{round}(\beta)$ and the numeric value of deviation is calculated as $\alpha = \beta - i$.

For example, in Figure 2.5, the linguistic term M (medium) has its numeric center equal to 3.0. In case of a value of 3.25 the "deviation" from M is 0.25. This approach allows for keeping all the original information – the translation takes place but no information is lost. The process of construing 2-tuples is presented more formally below.

So, the translation of value into its equivalent 2-tuple is done using the following way:

$$\Delta : [0, n] \rightarrow T \times [-0.5, 0.5)$$

$$\Delta(\beta) = \begin{cases} t_i \;\; i = round(\beta) \\ \alpha = \beta - i \; \alpha \in [-0.5, 0.5) \end{cases}$$

For example, the linguistic 2-tuple of the symbolic aggregation result β = 4.2 in a linguistic term set $T = \{t_0, t_1, t_2, t_3, t_4, t_5, t_6\}$ is represented by $\Delta(\beta) = (t_4, +0.2)$.

Chapter 3

3 Related work

A number of approaches have been presented addressing the problem of similarity computation [3, 9, 13, 17, 22, 42, 54, 74]. We categorize and discuss them according to the deployed methodology, as shown in Figure 3.1.

In lexicon and syntax-based similarity, meanings of words and their structure are taken into account, respectively. Structural methods are constituted by hierarchies and relations, where two elements are compared based on their positions in the ontology they belong to. Information-based approaches are based on the probability models of the entities. In contrary to the methods above, feature-based methods evaluate semantic knowledge of the concepts in ontology and LD. In general, similarity assessment is either performed by a single measure or a combination of multiple measures as in hybrid methods. Below, we discuss some of the well-accepted similarity measures in the literature from each category.

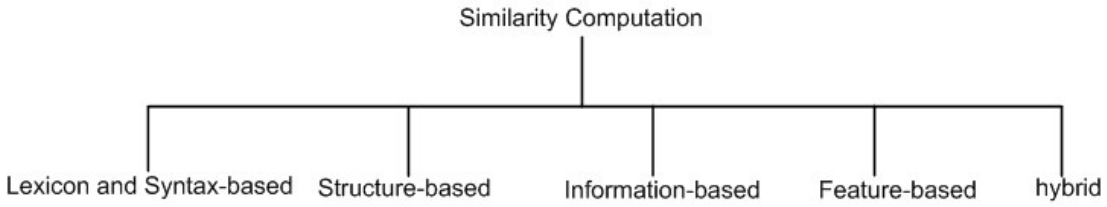


Figure 3.1 Similarity measurements classification

3.1 Existing similarity assessment techniques

3.1.1 Lexicon and syntax-based methods

Lexicon and syntax-based model leverages lexical and syntactical knowledge about the concepts, which is stored in external online repositories such as WordNet [47] as well as description of the text [22, 27, 32]**.** This model focuses on simple word matching without processing and understanding the concepts, thus ignores some important semantics factors.

Authors in [27, 40] introduce measures that evaluate the similarity of two concepts according to their literal values. The method presented in [40], known as edit distance, is based on the smallest number of insertions, deletions, and substitutions required to transform one concept to another. In the application of ontology mapping, [22] utilizes a series of string-based, sense-based and gloss-based similarity matching techniques in order to capture the similarity correspondences between concepts of two different ontologies. The lexical matching in [22] exploits synsets of words by using WordNet[22] as an external lexical database for English words. However, it is observed that some pairs of similar concepts are categorized dissimilar due to the bias of lexical similarity technique. Let us consider an example, letter “b” and number “2” lexically are dissimilar; however, they are semantically similar if they both represent section orders of a book.

Overall, in syntax-based methods the similarity measure is obtained by assessing the lexicographic information related to literal values of words (concepts), e.g., string matching

[22] http://wordnet.princeton.edu/

algorithms. As an enhancement, external knowledge repositories are adopted including WordNet [55]. Instead, in this report we focus on the underlying semantic similarity between the entities that cannot be found in syntax and lexicon.

3.1.2 Structure-based methods

The similarity measures presented and used in [17, 37, 61, 74] rely on structural information found in relationships existing between the concepts in ontology. In [61], conceptual distance is defined as the shortest path through a semantic network of is-a hierarchical relations between any two concepts. The approach used in [74] is purely based on the structure of the hierarchy; similarity between two ontology elements is defined as the number of nodes that separate the two concepts from the root node and the distance between the least common subsuming concept of the two entities to the root node. An extension of the work presented in [74] is reported in [67]. The authors improved the relevancy of similarity measure between concepts located in the same hierarchy. Likewise, the measure of similarity presented in [37] is based on the length of the shortest is-a path between the concepts and the maximum depth of ontology (node counting). These approaches for similarity measurement are simple to calculate but limited to the assumption of a tree or a lattice of ontology. Another problem with structure-based methods is that all hierarchical links between concepts are assumed to have a uniform weight. Hence, they do not represent a deep semantic relation. It is shown that structural knowledge does not correlate well to human judgment of similarity [62].

In order to enhance a loose notion of uniform structural correspondence, [17] proposes a similarity measurement that utilizes fuzzy set theory [77] to form fuzzy schema knowledge in

which the relationships between concepts are fuzzy numbers. In this framework, Associative Network (AN) is described as a semantic network consists of nodes as concepts and edges as relationships between the concepts. Edges are weighted with fuzzy numbers indicating the strength of belief in that relation. Similarity is calculated as the combination of path weights on the shortest distance between the two concepts using different functions of triangular norms from fuzzy set intersections. A drawback with this work and several others adopting this technique such as the one described in [69] is that the path weights are assumed to be given in advance by user or domain expert.

Below, we briefly describe number of approaches on the topic of similarity assessment that are using different techniques to measure the relatedness of concepts in ontology.

Path-based methods are based on the structure of taxonomic hierarchy in an ontology such as number of nodes (concepts) separating the two concepts [37, 61]. The method in [37] is based on the formula:

$$similarity = -\log(\frac{length}{2*D}) \tag{3.1}$$

where *length* is the length of the shortest path between two concepts, and *D* is the maximum depth of the used ontology. In [74] , similarity is computed based on the distance of the concepts and their common super concept to the root node:

$$similarity = \frac{2*depth(lcs)}{depth(e_1)+depth(e_2)} \tag{3.2}$$

where *lcs* is the least common super concept (subsume), and *depth(e)* is the depth of a concept *e* in the used concept structure. In general, considering the taxonomy into the similarity measure improves the result to be closer to human similarity judgment. Yet, it still lacks description of the semantic since it only uses the subsumption (*is-a*) relations between concepts.

In summary, in ontology schema-based measures the computed similarity is based on the structure of concepts and their relations in the ontology schema, see [14, 44]. The accuracy of the similarity value is heavily dependent to the quality of the human-designed ontology. This measure cannot properly calculate the similarity between concepts expressed in RDF triples and instances. These measures are primarily useful to be adopted in ontology alignment and mapping [16], where the main interest is finding the similarity associations between concepts. However, the similarity measures proposed in chapter 8 is applicable to RDF triples and instances while it also exploits the information contained in the ontology.

3.1.3 Information-based methods

In information-based measures, similarity is evaluated based on the amount of information about each entity, as introduced by [62]. The research work that fall under this model calculate relatedness of concepts using the available information about the concepts, which may be found in statistical knowledge and probabilistic model of the domain.

Different methods have various definitions for the amount of information such as the probability of occurrence of a concept in a huge reference corpus [65], combining information

from multiple corpora [41], and independent of parsing a corpus by calculating the cardinality of sub-concepts and total concepts as defined in the ontology [58].

The work in [42, 62] is based on measuring relatedness of objects using the amount of information about them. In [42], information content in commonality and dissimilarity of two objects is calculated using the probabilistic model of the domain. A method based on both information contents and taxonomic structure of objects is proposed in [62]. The problem with these approaches is the need for a complete statistical view of information in order to determine the relatedness of two entities. Moreover, information-based similarity methods are highly dependent to the appropriate choose of word senses.

In [42], the matching is based on the amount of information needed to describe the commonality of concepts as well as information that describes each concept. In [62], semantic similarity is quantified using shared amount of information between two concepts, which is indicated by the information content of the concepts that subsume them in the taxonomy. The formula used in [62] is:

$$similarity = \max_{e \in S(e_1, e_2)} \left[-\log p(e) \right] \tag{3.3}$$

where $p(x)$ is the probability of encountering an instance of concept x. $S(e_1, e_2)$ is the set of subsuming concepts of the concepts e_1 and e_2. The defined similarity measure in information-theoretic model is usually dependent to an existence of probabilistic information about the concepts in ontology, such as frequency of concepts occurrences in the taxonomy, external corpora, or informativeness of relations between concepts.

3.1.4 Feature-based methods

Tversky introduced this model in [72] and several approaches have been proposed adopting this model and combining it with other methods. The simplest model of feature-based similarity approach is obtained by counting the number of common and distinctive object properties of a concept, which is formally introduced in [72]. In [72], psychological validity of symmetric similarity and triangular inequality are argued.

Methods based on this model are different from the other approaches by relying on intrinsic relationship between the concepts extracted. Started by Tversky's formulation of similarity [72] where a similarity degree is determined based on the features of concepts. The formula used for this purpose is:

$$similarity = \frac{|X \cap Y|}{|X \cup Y| + |X - Y| + |Y - X|} \tag{3.4}$$

where X and Y are sets of features for each of the concepts, and |.| is cardinality of a set.

A degree of similarity is determined using the modified version of the Tversky's index, known as Dice index [20]:

$$similarity = \frac{2 * |X \cap Y|}{|X| + |Y|} \tag{3.5}$$

In [57], feature-based model of similarity is combined with information theoretic model in which information content of features are also taken into account. In [57], quantification of information content of features is presented as an extension to the traditional information

theoretical models, as presented in [42] that uses corpora analysis. A limitation of [57] is that it can measure the similarity of concepts in ontology, where number of concepts and subconcepts can be determined. However, LD is a wide network that consists of several datasets in which calculation of total number of concepts is not feasible.

[46] defines an iterative similarity identification measure between concepts in LD by considering the information content of common set of features between a pair of concepts as well as features that reside outside of this set. [46] also discusses weighted similarity by considering pre-defined weights assigned to specific link types to illustrate their importance. The weights are assumed to be pre-defined and given in advance. In another work, similarity in [2] is scaled up to LD while different problems in similarity assessment of concepts in LD are discussed and possible solutions to them are investigated including non-authoritative data, inconsistent data and computational problems.

On the application of information retrieval and integration, [63] aims at determining semantic similarity in ontology between entities based on the similarity combination of lexicographic features, synonym sets, and semantic neighborhoods. For each component, Tversky's formulation of similarity is adopted while equal weights are assigned to each similarity component. In [63] direction of asymmetric similarity between two concepts is determined according to their degree of granularity in the corresponding ontology. However, relations between two concepts are limited to is-a and part-of relations and are extracted from WordNet.

The approach proposed in [24] measures the similarity between symbolic objects by considering the information about quantitative and qualitative features of the objects. This information includes relative position of feature values (applicable only to quantitative components), relative sizes of feature values, and a measure representing common parts between these feature values. The work described in [9] is an application of this approach. It presents a divisive hierarchical clustering algorithm that makes use of both symbolic similarity and dissimilarity measures. Similarity and dissimilarity measures are computed using the method in [24]. A problem with this similarity measure is that sine and cosine measurements are applicable only when objects are represented as numerical feature vectors.

In this category, RDF-based measures are the approaches in which similarity is assessed by comparing the RDF triples, representing features, associated with each entity, see [33, 34]. The methodologies proposed in chapters 6-9 follow this approach. In Semantic Web entities descriptions are disseminated in form of RDF triples, thus they turn out to be useful for evaluating the semantic similarity between entities. Moreover, calculating the similarity through evaluating RDF triples provides integrated facts via extra links to other datasets.

Among all the above models, feature-based model is shown to be an efficient similarity technique to be applied in the framework of ontology and RDF, and it is successfully applied in many applications [15, 23, 56, 70]. The underlying reason is that the result from feature-based model is very close to human perception of similarity. Our proposed approaches in chapters 6-9 are motivated by feature-based model.

3.1.5 Hybrid methods

In a number of cases similarity measurement processes are done by combining single measures [13, 18, 69]. The matching process proposed in [69] introduces syntactic and semantic matching to perform matchmaking for agent advertisements and requests in Internet. They apply five different similarity measures that act as filters to determine a set of matching results such as context matching and comparison of agent's profiles. Based on the required matching degree, different combinations of these similarity filters are applied. Semantic distance between concepts is measured with regards to subsumption relationships and additional associations using a weighted associative network [17]. One drawback is that relations between concepts are labeled with a difficult to be obtained weights. Among similarity measures shown in [69] are syntactic distances between keywords and term-frequency-inverse document weighting.

A system for combining matching algorithms (COMA) [13] and its extension COMA++ [3] represents a schema matching system as a platform to combine multiple matchers in a flexible way. They assume a directed acyclic graph representation of schemas. Multiple matchers that are stored in a matching library operate independently, and each determines a similarity value. The result is a set of mapping entities with similarity values between 0 (strong dissimilarity) and 1 (strong similarity) provided by each matcher, which creates a cube of similarity values. A single similarity value is obtained by aggregating the results of all the matchers using average and Dice coefficient. The implemented matchers in the library support syntactical (n-grams, edit distance, affix, and soundex matching), lexical (datatype and label matching), and structural

properties of a schema. However, the semantic behind the concepts in ontology, represented by RDF interconnections, is not incorporated in the similarity measurement process.

In [18], lexical, structural and feature-based information is contributing in the similarity measurement between ontologies. Immediate features are defined based on the concept roles, where objects in RDF triples are taken as concept's roles and are being compared. Matching techniques are applied in a layered fashion; the list of matching concepts is shortlisted and only those concepts that met the similarity criteria in the last phase(s) are passed on to the next layer. The first drawback of this approach is the questionable independency of similarity criteria in each layer, e.g. according to [18], two concepts are determined to be dissimilar when they are taxonomically (structure-based) similar but not lexically. We argue that similarity measurement techniques should be applied as a combination and not in a layered fashion. Second, the structural and feature-based measures look for exact matches. However, this exactness does not accommodate well with the definition of similarity.

3.2 Other research work

Among several literatures available on similarity in LD [5, 12, 64], we focus on the work related to our approach in terms of the proposed method to compute the similarity between entities in LD and the nature of data under evaluation. For extensive review of current semantic similarity techniques see [11].

The work presented in chapter 9 revolves around semantic matching in the application of pharmacy by matching drugs to a referenced drug using fuzzy linguistic representation model.

Some of the related research work in these areas is discussed here. The topic of nonprescription drugs and their selection is the objective of the web application in [10], winner of the third place in 2012 Semantic Web Challenge [28]. The approach is based on collecting data from LOD datasets relevant to the domain of medication. [10] supports our claim regarding the lack of drug-related information that can be retrieved from LD. This approach for semantic processing of information is based on identifying drugs' molecules with the WHO[23] ATC classification as well as their own developed ontology. In this direction, the key properties of a drug are detected and contributed to the designed ontology. In general, designing an ontology and using it in the process of similarity calculation limits the scope of the real-data to a human based taxonomy and ties the semantic similarity to the distance between concepts in ontology. Details of computing the semantic similarity is not explained which has made us unable to compare it with our method.

Concerning fuzzy linguistic representation model applied in chapter 9, [48] presents a method for accessing relevant information in the domain of digital libraries. [48] obtains relevance of user profiles to resources and other profiles for digital libraries applying the 2-tuple based linguistic representation model. Interestingly, the degree of interest of the user about a particular topic is represented by fuzzy linguistic labels that are used to generate 2-tuple linguistic labels. For computing the relevance of concepts, authors rely on an external information repository namely hierarchical taxonomy of the system as proposed in [52]. In [52], similarity is computed based on the position of two concepts within the taxonomy and the

[23] http://www.who.int/en/

deepest hierarchy level. As previously mentioned, the problem of ontology distance is its dependency on the construction of the taxonomy, which is a highly subjective engineering task. However, in our semantic-based approach in chapter 9, the similarity between drugs is determined without a need for external information.

In [21] a query mechanism is built on the basis of semantic similarity between query terms and concepts in LD. In [21], semantic similarity is referred to as semantic relatedness to express independency to any taxonomic vocabularies. The semantic relatedness between query words and concepts in LD is measured based on the principle of distributional semantic [71]. The underlying idea in distributional semantic is to use statistical distribution of word co-occurrence in texts as the semantic representation of words. Matching process between query terms and vocabulary is performed over properties, types and instances of each term as described in RDF triples. However, similarity is calculated based on the link structure between Wikipedia[24] pages of the terms. In this process, each link in the corpora is assigned a weight according to term frequency–inverse document frequency, TF/IDF measure. Similarity between two terms is calculated based on the number of articles containing each term and total number of articles in Wikipedia. Using a text-based dataset, Wikipedia, limits the scope of the similarity within boundaries of traditional web pages. In our method in chapters 6-9, the basis of similarity calculation is RDF triples, which reflect the full semantic description of concepts.

Query engines for Semantic Web have been proposed in multiple papers [4, 29, 60]. The authors of [60] present a query engine called DARQ to address the problem of distributed RDF

[24] http://en.wikipedia.org/

data on the web and its integration. This query engine is compatible with any SPARQL endpoints and it allows updates to the list querying datasets without user intervention. [60] shows an improvement in query performance compared to SPARQL results by introducing a query optimization algorithm. DBpeida[25] data set was used for evaluation and measures of query execution time and transformation time are reported. Authors discuss the importance of dealing with information representation from multiple data sources and plan to adjust the query patterns accordingly. In this subject, we believe that using linguistic information and calculations will greatly benefit engines' performance in retrieving relevant items.

Majority of today's methods are not proper to be applied in LD as a similarity computation technique. Note that taxonomies of different datasets are not comparable therefore ontology-based approaches encounter problems when concepts belong to different datasets in LD. Corpus-based methods have shown reasonable alignment with human judgment of similarity however these techniques focus on traditional information representation models such as web pages and documents and are a poor way to capture the semantic of terms. In fact, the amount of available RDF data on the web is very big and heterogeneous compared to the used vocabularies and schemas. We argue that similarity methods that take features of a concept into consideration are best suited for the vast network of connected data, LD.

[25] http://dbpedia.org/About

Chapter 4

4 A new approach to semantic similarity evaluation of concepts defined in ontology[26]

In this chapter, the proposed method aims for determining semantic similarity between concepts defined in ontology. In contrast to other techniques that use ontological definition of concepts for similarity assessment, the proposed approach focuses on the relations between concepts and their semantics. In addition, the method allows for context-aware similarity assessment as well as similarity discovery between instances of concepts.

4.1 Semantic similarity evaluation approach

Let us define ontology as a 5-tuple:

$$O = \{C, R, H^C, rel, A^O\} \tag{4.1}$$

where *C* is a set of concepts and data types, *R* is a set of non-taxonomical relations, i.e., all relations including object properties and datatype properties excluding *is-a*, H^C is a concept hierarchy (taxonomy), where $H^C(c_1, c_2)$ means that c_1 is a subconcept of c_2, a function $rel: R \rightarrow C \times C$ that relates non-taxonomical relations to concepts, i.e., $rel(R) = (c_1, c_2)$ is equivalent to $R(c_1, c_2)$ that is a set of relations of type *R* between c_i and c_j . Also, A^O is a set of ontology

[26] P. D. Hossein Zadeh and M. Z. Reformat. (2013) Assessment of semantic similarity of concepts defined in ontology. International Journal of Information Sciences, Elsevier. volume 250. pp: 21-39. (published)

axioms. An instance of a concept c_i – called an individual – is denoted as $c_i.ins_m$, where the subscript *m* means that a single concept can have multiple instances.

The overall similarity between two concepts is introduced in two components. The first component – sim_1 – represents a contribution to the similarity between c_i and c_j and is determined based on direct relations between c_i and c_j, as well as common features shared between both concepts:

$$sim_1(c_i, c_j) = \left|R(c_i, c_j)\right| + \sum_{c_k \in N(ij)} \left[\max_{\substack{r_i \in R(c_i, c_k) \\ r_j \in R(c_j, c_k)}} \left\{ relationSim(r_i, r_j) \right\} \right] \tag{4.2}$$

where *N(i)* denotes a set of concepts that concept c_i is connected to in a given ontology, and $N(ij)=N_{common}(c_i,c_j)$ is the set of common concepts that both c_i and c_j are connected. Also, $\left|R(c_i, c_j)\right|$ is the number of direct connections between two concepts c_i and c_j, while $\left|.\right|$ represents cardinality of a set. *relationSim()* represents a function that evaluates similarity between relations. This function can be built based on any approach using structure-, lexicon-, or string-based similarity measures. In this work, we used the structure-based method defined in [74] to measure the similarity of relations.

The second component – sim_2 – measures contributions to the similarity emerging from features that are unique to each concept c_i and c_j. It should be noted that $N_i^o = N(i) - N(ij)$ and $N_j^o = N(j) - N(ij)$ represent unique features of each concept c_i and c_j, respectively.

$$sim_2(c_i, c_j) = \sum_{c_z \in N^o(i)} \left[\max_{\substack{c_y \in N^o(j) \\ r_i \in R(c_i, c_z) \\ r_j \in R(c_j, c_y)}} \{relationSim(r_i, r_j)\} \oplus \max_{w \in Y} \{sim(c_z, c_w)\} \right] \quad (4.3)$$

where the set *Y* is obtained as $Y = \underset{c_y}{\text{argmax}} \{relationSim(r_i, r_j)\}$, for a given c_z and concepts c_y from $N^o(j)$.

The set *Y* represents a set of concepts that belong to $N^o(j)$ and connected to c_j via relations which are the most similar to $R(c_i, c_z)$. Note that $\oplus$ is an aggregation function, which we use a t-norm function taken from fuzzy set theory [77].

Finally, the similarity between concepts c_i and c_j is defined as:

$$sim(c_i, c_j) = \frac{sim_1(c_i, c_j) + sim_2(c_i, c_j)}{|N(i)|} \quad (4.4)$$

As can be seen, the obtained similarity in Eq. (4.4) is asymmetric.

So far, we have focused on concepts of ontology. However, we can also consider individuals that are instances of concepts. The formula for similarity of individuals, *simInd*, is expressed as:

$$simInd(c_i.ins, c_j.ins) = \frac{simInd_1(c_i.ins, c_j.ins) + simInd_2(c_i.ins, c_j.ins)}{|N(i)|} \quad (4.5)$$

where,

$$simInd_1(c_i.ins, c_j.ins) = \left|R(c_i.ins, c_j.ins)\right| + \sum_{c_k.ins \in N_{ins}(ij)} \left[\max_{\substack{r_i \in R(c_i, c_k) \\ r_j \in R(c_j, c_k)}} (relationSim(r_i, r_j)) \right] \tag{4.6}$$

c.ins indicates an instance of a concept *c*. Also, $simInd_2$ is as follows:

$$simInd_2(c_i.ins, c_j.ins) = \sum_{c_z.ins \in N_{ins}{}^{o}(i)} \left[\max_{\substack{c_y.ins \in N_{ins}{}^{o}(j) \\ r_i \in R(c_i, c_z) \\ r_j \in R(c_j, c_y)}} \left\{ relationSim(r_i, r_j) \right\} \oplus \max_{w \in Y} \left\{ sim^{IND}(c_z.ins, c_w.ins) \right\} \right] \tag{4.7}$$

with

$$sim^{IND}(c_i.ins, c_j.ins) = \begin{cases} simInd(c_i.ins, c_j.ins) & if\ c_i, c_j\ are\ concepts \\ stringmatch(c_i.ins, c_j.ins) & if\ c_i, c_j\ are\ datatypes \end{cases} \tag{4.8}$$

$N_{ins}(ij)$ represents instances shared between $c_i.ins$ and $c_j.ins$, and $N^{o}_{ins}(i)$ is a set of instances unique for $c_i.ins$. The relations r_i and r_j between instances of concepts are defined at an abstract level, $R(c_i, c_z)$ and $R(c_j, c_y)$.

In order to determine similarity between two concepts c_i and c_j under a specified context (cntx), we need to define some quantities. Let $N^{cntx}(l)$ denote a set of concepts and literals that concept c_l is connected to via relations that belong to the context defined by a set of relations R^{cntx}. $N^{cntx}(ij) = N^{cntx}_{common}(c_i, c_j)$ is the set of common concepts that both c_i and c_j are connected to them via relations that belong to the context. With such definitions $sim_1^{cntx}(c_i, c_j)$ is reformulated as:

$$sim_1^{cntx}(c_i, c_j) = \left| R^{cntx}(c_i, c_j) \right| + \sum_{c_k \in N(ij)} \left[\max_{\substack{r_i \in R^{cntx}(c_i, c_k) \\ r_j \in R^{cntx}(c_j, c_k)}} (relationSim(r_i, r_j)) \right] \tag{4.9}$$

Also, $sim_2^{cntx}(c_i, c_j)$ is obtained as:

$$sim_2^{cntx}(c_i, c_j) = \sum_{c_z \in N^{o,cntx}(i)} \left[\max_{\substack{c_y \in N^{o,cntx}(j) \\ r_i \in R^{cntx}(c_i, c_z) \\ r_j \in R^{cntx}(c_j, c_y)}} \left\{ relationSim(r_i, r_j) \right\} \oplus \max_{w \in Y} \left\{ sim^{cntx}(c_z, c_w) \right\} \right] \tag{4.10}$$

where set *Y* is defined as $Y = \underset{c_y}{\text{argmax}} \left\{ relationSim(r_i, r_j) \right\}$, and $\oplus$ is a t-norm function.

Finally, the context-aware similarity between concepts c_i and c_j is defined as:

$$sim^{cntx}(c_i, c_j) = \frac{sim_1^{cntx}(c_i, c_j) + sim_2^{cntx}(c_i, c_j)}{\left| N^{cntx}(i) \right|} \tag{4.11}$$

Eq. (4.9-4.11) consider only those connections that define the context when compared to Eq. (4.2-4.8).

4.2 Experiments and comparison studies

The prototypical ontology used here is built by importing, modifying and extending three already existing ontologies: ontology of concepts related to academic research[27], ontology of

27 ka.owl: http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library#OWL_ontologies

university concepts[28], and ontology of computing science concepts[29]. An open source ontology editor Protégé 4.2 beta [25] has been used. A snapshot of the developed ontology is shown in Figure 4.1.

In order to compare our proposed approach to other well-known techniques in the literature, an experiment has been designed. A set of pairs is selected from the defined ontology, as shown in Table 4.1. We gathered human judgment of similarity in order to assess the similarity values obtained from the methods. For this reason, a special portal using SurveyMonkey[30] services is prepared. We have asked 100 individuals from different backgrounds to provide their similarity estimation for each pair of concepts and give us a value between 0 (if the concepts in a pair are dissimilar) and 1 (if the concepts are perfectly similar). Two versions of the experiment is conducted: 1 – each pair of concepts is shown along with a list of features for each concept based on our ontology, 2 – pairs of concepts are presented without any additional information. The standard deviation ranges for similarity scores calculated for each pair are [0.25, 0.32] for version 1, and [0.25, 0.33] for version 2 of the experiment.

[28] univ-bench.owl: http://swat.cse.lehigh.edu/projects/lubm/

[29] http://www.owl-ontologies.com/ComputingOntology.owl

[30] http://www.surveymonkey.com

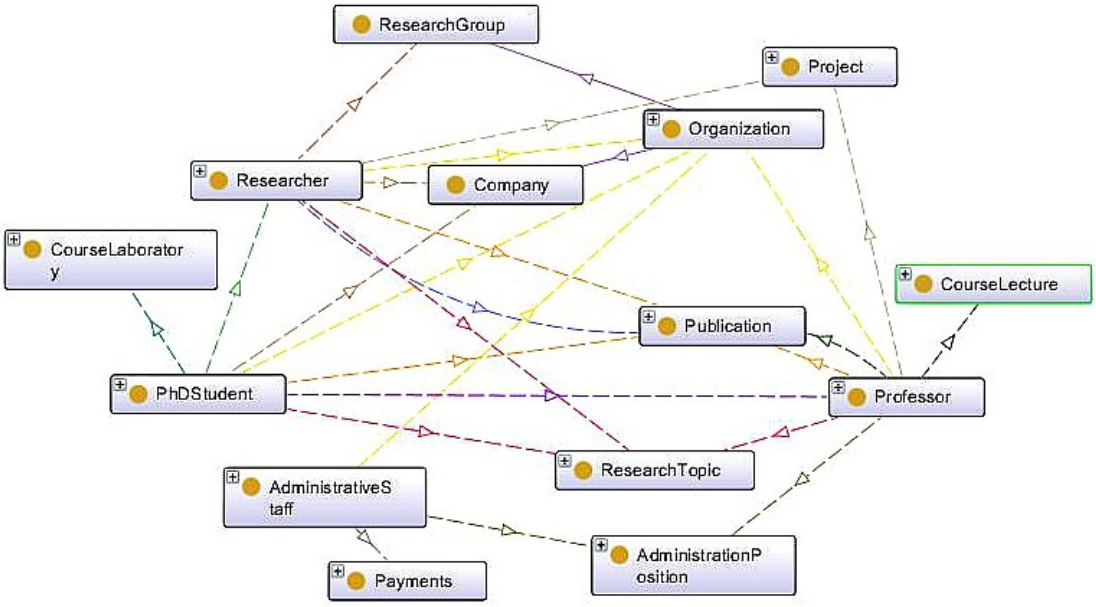


Figure 4.1 Four concepts of *Professor*, *Researcher*, *PhDStudent*, and (*Admin)istration Staff* and their connections. Arcs denote different relations between the concepts

According to Table 4.1, our method performed relatively well compared to other methods. As can be seen, our method and other feature-based methods overestimate the similarity for the pair {*professor*, *phd student*}. However, the similarity values for this pair given by the two structure-based methods, Wu & Palmer [74] and Leacock & Chodorow [37], are closest to the human judgment. In general, it can be observed that for all pairs the feature-based methods outperform other techniques. This can be explained via the semantic-oriented nature of feature-based methods. Overall, the similarity estimations provided by our method are closer to human judgment values obtained in version 1 than version 2. Therefore, we may claim that our method reaches reasonably good results when the constructed ontology is rich and well-defined in the domain of knowledge.

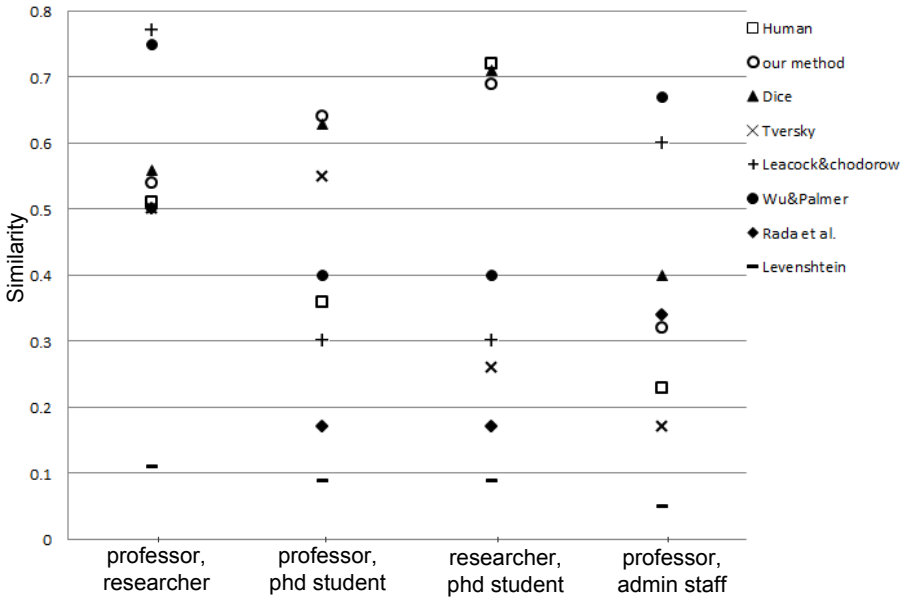
Table 4.1 Similarity values for multiple similarity assessment methods

Concept pairs	Levenshtein metric [40]	Rada et al. method [61]	Leacock & Chodorow method [37]	Wu & Palmer method [74]	Tversky index [72]	Dice index [20]	Our method	Human judgment		
								(1)	(2)	Avg.
professor, researcher	0.11	0.50	0.77	0.75	0.50	0.56	0.54	0.55	0.47	0.51
professor, phd student	0.09	0.17	0.30	0.40	0.55	0.63	0.64	0.42	0.30	0.36
researcher, phd student	0.09	0.17	0.30	0.40	0.26	0.71	0.69	0.70	0.74	0.72
professor, admin staff	0.05	0.34	0.60	0.67	0.17	0.40	0.32	0.21	0.25	0.23

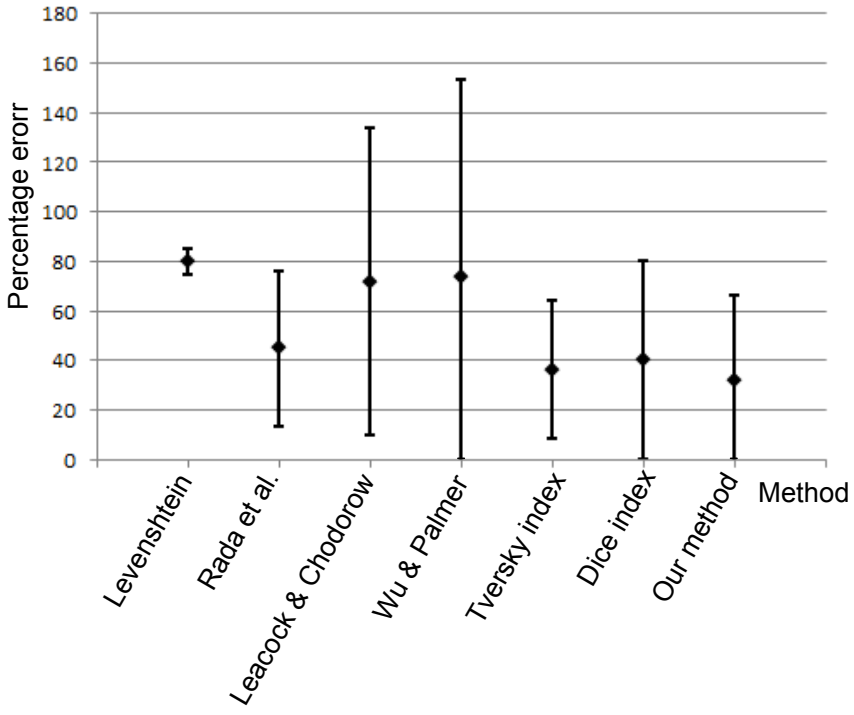
The large number of participants allowed us to do some statistical analysis of the obtained results. We used human judgments as reference points (ground truth) and calculated percentage error for each similarity assessment method with the following formula:

$$percentage\_error(method_i) = \left(\frac{\left| human\_avg - similarity\_method_i \right|}{human\_avg} \right) \times 100 \qquad (4.12)$$

where *similarity_method*$_i$ represents the similarity value obtained by the method *i* as shown in Table 4.1, and *human_avg* as the average value of human judgments over versions 1 and 2. Figure 4.2.a illustrates the scatter plot of similarity values of all methods, as well as average human judgment similarity, *human_avg*, for each pair. Figure 4.2.b shows the average percentage error over all pairs for each method, using Eq. (4.12). The confidence intervals are obtained based on the mean and standard deviation of the computed errors.


Similarity
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
professor, researcher
professor, phd student
researcher, phd student
professor, admin staff
Human
our method
Dice
Tversky
Leacock&chodorow
Wu&Palmer
Rada et al.
Levenshtein

(b)


Percentage erorr
180
160
140
120
100
80
60
40
20
0
Levenshtein
Rada et al.
Leacock & Chodorow
Wu & Palmer
Tversky index
Dice index
Our method
Method

(a)

Figure 4.2 Comparison of similarity values of each method for each pair (a), and average percentage error over all pairs for each method (b)

The results indicate that our method has the smallest average error compared with other techniques in this study. This can be explained as our method takes into account the underlying semantics of concept's features. Moreover, our method not only investigates the immediate features of concepts but it also evaluates the features located further away by recursively traversing the ontology. We have performed the paired t-test with a desired critical t-value of 80% confidence and the degree of freedom of 3. It has shown that our method's results are statistically significant than results obtained by all methods except for Rada et al. [61], Dice [20] and Tversky's index [72]. Although our method's average error is smaller than the average error by Rada et al., Dice and Tversky index the difference is not statistically significant.

Chapter 5

5 Feature-based similarity assessment in ontology using fuzzy set theory[31]

With growing number of web pages and available data on the Web the need has arisen for development of effective tools to manage and facilitate access to information stored on the Web. In this process, one of the challenges is finding the most relevant data to the user's interest. In today's Web, relevancy or in other words similarity is evaluated by keyword matching of the query to the pieces of information on the Web. In Semantic Web [38], as a new paradigm providing a novel vision for data representation on the Web, information is presented within a conceptualization hierarchy referred to as *ontology*. Ontology is expressed by a formal ontology language as Web Ontology Language (OWL). The language is implemented based on information triples defined in the context of Resource Description Framework (RDF) [51]. Thus, in Semantic Web similarity may be computed using the semantics of concepts in ontology. In fact, evaluating the similarity is a central component of a number of tasks performed in Semantic Web, such as data-mining, reasoning, search engines, information retrieval, clustering, ontology mapping, and ontology translation, see [59, 73].

In Semantic Web, every piece of information is presented with RDF triples, which provides a user/machine understandable meaning to every concept. In this chapter, the proposed

[31] P. D. Hossein Zadeh and M. Z. Reformat. (2012) Feature-based Similarity Assessment in Ontology using Fuzzy Set Theory. IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). pp: 1-7. (published)

approach determines the semantic similarity in the framework of ontology and RDF data model based on the relationships between concepts. In particular, *semantic* similarity between concepts is computed by analyzing their interconnections regardless of whether they reside within the same ontology or not. In other words, we aim to determine the similarity by considering all kinds of relations between two concepts including hierarchical (*is-a*) and non-hierarchical. Not only hierarchical *is-a* relations are considered but also all types of relations between the concepts. This enables a full range of possible relations, which carries semantics about the concepts under study, to be involved in the similarity measure [23, 45]. Fuzzy set theory [77] is used to quantify the similarity measure at different levels of abstraction in ontology. Furthermore, selection of different types of interconnections according to the defined criteria allows the *context* to be involved in the similarity assessment. This means determining similarity measures based on a selected subset of connections between two concepts according to the given context.

5.1 Fuzzy semantic matching technique

In this section, we propose a technique for determining semantic similarity between pieces of information defined in ontology based on features describing each piece of information. The presented method allows for considering a specific context into the similarity evaluation. The quantitative characterization of similarity at different levels of abstraction in ontology is provided using elements of fuzzy set theory. In section 5.2, we show through experiments that the proposed method compares favorably to other measures in terms of human judgment of similarity.

In this method, similarity of two concepts is determined according to the connections between them and the connections that both concepts share to the same other concepts in ontology. In fact, features of a concept are expressed through the connections of that concept in ontology.

To accommodate the changes in requirements of the methodology in this chapter, we adapted the ontology definition in (4.1) of Chapter 4 to a 3-tuple:

$$O = \{E, R, f\} \tag{5.1}$$

where *E* is a set of concepts (entities) in ontology *O*, *R* is a set of connections between concepts, and *f* is a function that states if any two concepts are connected or not, i.e., $f(e_1,e_2)$.

In real-life scenarios, it is common to assess the similarity between concepts in a specific context. It is demonstrated that context plays an important role in semantic of concepts [54]. In such a case, only a subset of features is taken into the similarity evaluation. In ontology, the context is defined by a single or a set of properties. For example, similarity between a professor and a PhD Student can be analyzed in the context of their research interests, or published papers. Considering each of these contexts is equivalent to evaluating the relevant defined properties in the ontology, e.g., properties "ResearchInterests" and "Published", respectively.

The semantic similarity between two concepts *A* and *B* is defined as follows:

$$sim(A, B \vdots R_c) = \frac{n_{cm}(A, B \vdots R_c)}{n(A \vdots R_c)} \tag{5.2}$$

R_c is a set of connections ($R_c \subset R$) defined by the context. $n(A:R_c)$ denotes the number of features (connections) of concept A within the context *C*. $n_{cm}(A,B:R_c)$ is the number of common features between concepts A and B defined within the context *C*. The context C imposes some constraints on selection of the concept relations.

In a generic scenario of evaluating similarity of two concepts *A* and *B* without a specific context, Eq. (5.2) converts into:

$$sim(A,B) = \frac{n_{cm}(A,B)}{n(A)} \tag{5.3}$$

where the similarity is the ratio of total number of features in common for concepts *A* and *B* to the total number of features of concept *A*. It is worth noting that the similarity measure is asymmetric here. This is due to the existence of different number of features defining each concept. There exist several empirical evidences with regards to the presence of asymmetric similarity. In [72], it is shown that similarity should not be treated in a symmetric fashion while the direction of asymmetry is dependent on the prominence of the concepts.

Recall that different levels of abstraction in ontology influence the conceptual distances between the concepts. In particular, the farther one travels down in an ontology the conceptual distances decrease. The aforementioned conceptual distance can be observed throughout every ontology level; however, without losing the generality we focus only on the conceptual distances at definition and instance levels of an ontology. For example, the similarity between

the items "dolphin[32]" and "shark" (located in the instance level) is intuitively higher than the similarity of their associated super concepts found in the definition level, i.e., "mammal" and "fish", respectively. The similarity defined in Eq. (5.2) uses the level of ontology that the concepts reside in to generate a similarity value.

In order to find a reasonable similarity value we utilize fuzzy set theory based on Zadeh's definition [77]. Let *U* denote a universal set of similarity values given by Eq. (5.2). Based on two noticeable levels in ontology: "definition" and "instance", we define two types of similarity – a definition level similarity sim_{def}, and an instance level similarity sim_{ins}. For each of them, we define normalized fuzzy sets *low* and *high* for low and high values of similarities, respectively. In general, any number of fuzzy sets can be defined and used. The membership function μ_x for each fuzzy set *X* is defined in the standard way as:

$$\mu_X : U \rightarrow [0,1] \tag{5.4}$$

where [0, 1] denotes the interval of real numbers from 0 to 1 inclusive. Once the similarity value is calculated according to Eq. (5.2), then it is evaluated in terms of its degrees of membership in the defined fuzzy sets. Exemplary membership functions for the fuzzy sets *low* and *high* are determined based on the results of our preliminary experiments, Figure 5.1.a and 5.1.b. The values for the defined membership functions can be obtained through empirical studies depending on the context of application or in the process of personalization. As stated

[32] In this example, "dolphin" and "shark" are assumed to be actual instances in the ontology, and should not be confused to be concepts containing other sub concepts.

earlier, concepts at the definition level are more abstract when compared to the concepts at the instance level. Thus, higher similarity value is needed to indicate high similarity.

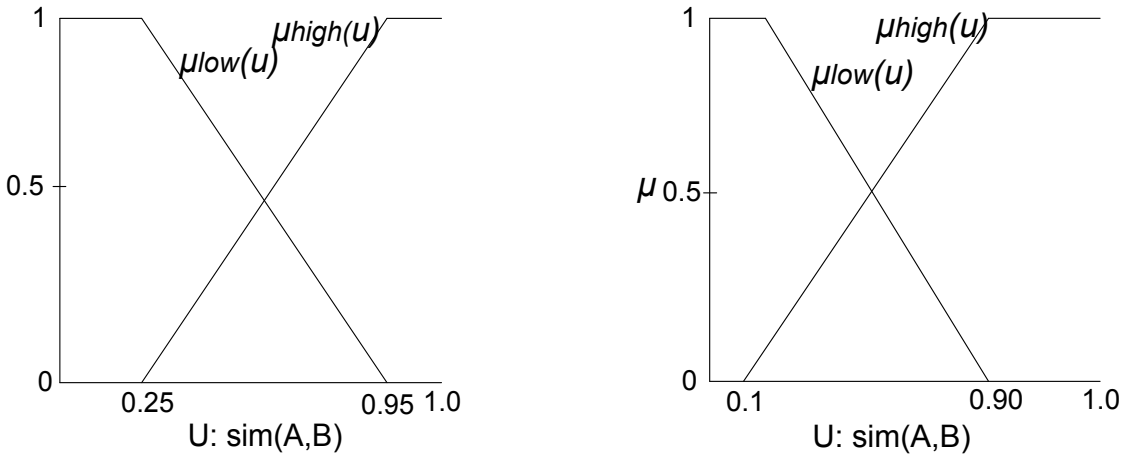


Figure 5.1 Membership functions of similarity at the definition level (a), and the instance level (b).

On the contrary, the similarity value required for implying similarity of concepts at the instance level is more relaxed. For example, at the definition level any similarity value below 0.25 is "labeled" as *low*, while any measure above 0.25 is *high* to some degree. At the instance level, any value above 0.10 is "labeled" as *high* to a degree.

For a given pair of concepts *(A,B)*, we find their similarity $sim(A,B:R_c)$ using Eq. (5.2). According to the level of ontology (i.e., definition or instance) that the concepts *(A,B)* are associated with, membership degrees of *sim(A,B)* to fuzzy sets *low* and *high* are obtained, which are indicated by $sim_{def}(A,B:R_c)$ and $sim_{ins}(A,B:R_c)$. It worth mentioning that level of belongingness for each concept can be determined from the syntax of ontology language and the annotations used in the ontology. For example, using Figure 5.1.a and 5.1.b and the

similarity value 0.3, calculated using Eq. (5.2), the degrees of membership of the similarity in the *low* and *high* fuzzy sets are shown in Table 5.1.

Table 5.1 Membership degrees of *sim(A,B)*

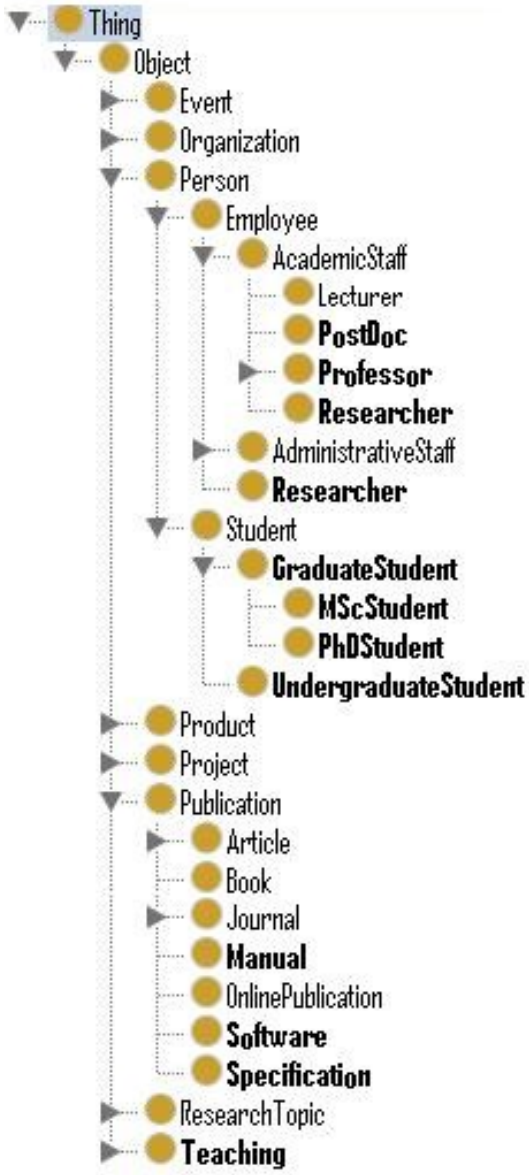
$sim(A,B:R_c)$	Levels	$\mu_{low}(u)$	$\mu_{high}(u)$
0.3	$sim_{def}(A,B:R_c)$	0.93	0.07
	$sim_{ins}(A,B:R_c)$	0.75	0.25

There are number of advantages of applying fuzzy approach in the process of similarity assessment. Firstly, it is simple and intuitive. Secondly, it gives a more human linguistic description of similarity judgment. Thirdly and most importantly, the values for *low* and *high* fuzzy sets, or any number of fuzzy sets, are easily customized to the needs of the user.

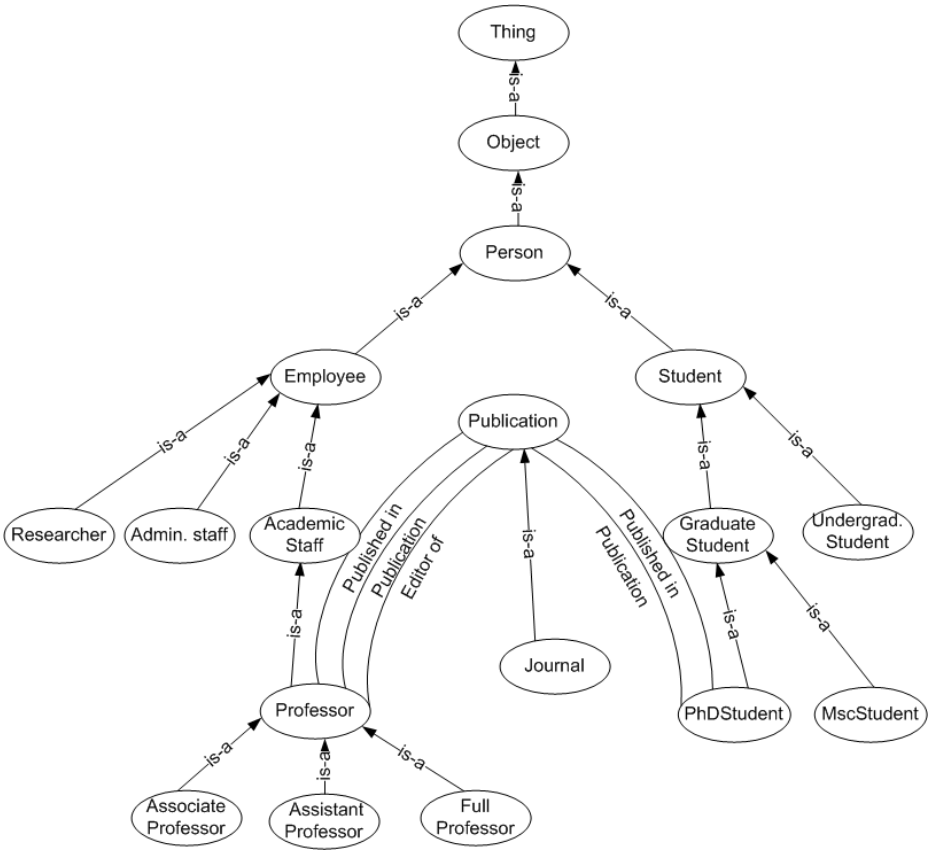
5.2 Experiments and comparison

The presented ontology in this section is an integration and modification of three existing ontologies: ontology of concepts from academic research[33], ontology of university concepts[34], and computing science concepts[35]. In the developed ontology, two concepts of *professor* and *PhDStudent* are defined as well as their properties, including connections to other concepts at definition level and their links to instances at the instance level of ontology. A snapshot the definition level of the ontology is shown in Figure 5.2.a and Figure 5.2.b.

[33] ka.owl: http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library#OWL_ontologies
[34] univ-bench.owl: http://swat.cse.lehigh.edu/projects/lubm/
[35] http://www.owl-ontologies.com/ComputingOntology.owl


Thing
Object
Event
Organization
Person
Employee
AcademicStaff
Lecturer
PostDoc
Professor
Researcher
AdministrativeStaff
Researcher
Student
GraduateStudent
MScStudent
PhDStudent
UndergraduateStudent
Product
Project
Publication
Article
Book
Journal
Manual
OnlinePublication
Software
Specification
ResearchTopic
Teaching

(a)



(b)

Figure 5.2 A snapshot (a) and a fragment (b) of the developed ontology

Figure 5.3 depicts a small fragment of the ontology including each particular concept with its associated properties. The concept *professor* has three connections (relations) to the other concept *Publication* while *PhDStudent* has two properties associated with *Publication*. At the instance level, it can be seen that there exist an instance of each concept *professor* and *PhD Student*: *professorX* and *PhDStudentY*. *ProfessorX* has 21 instances of published journal papers, and *PhDStudentY* has published 18 journal papers in total from which 8 of them are shared with the concept *professorX*. In Figure 5.3, the titles of the journal papers are indicated by "P#".

Let us calculate the similarity of the pair of concepts *professor* and *PhDStudent* at the definition level as well as the instance level according to the proposed approach. Based on the number of features for each concept as shown in Figure 5.3, and assuming that the set of defined properties in the context of publication is *{Published in, Publication, Editor of},* the similarity of the pair *(Professor, PhDStudent)* at the definition level in the context of publication is calculated in the following way:

$$sim_{def}(professor, PhDStudent : Publicatio\,n)$$
$$= \frac{n_{cm}(professor, PhDStudent : Publicatio\,n)}{n(professor : Publicatio\,n)} = \frac{2}{3} = 0.66$$

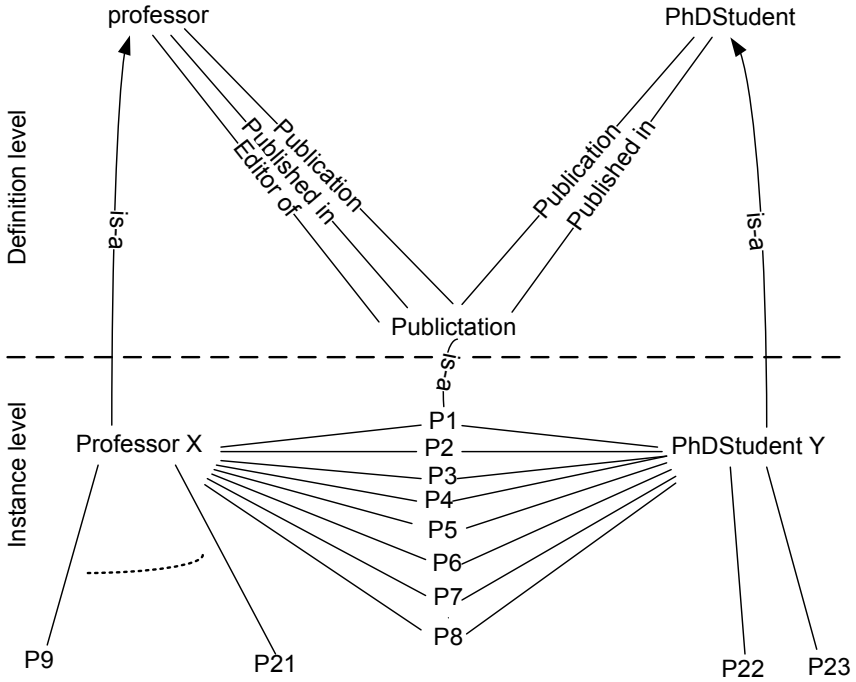


Figure 5.3 Concepts professor and PhDStudent in definition level and their instances ProfessorX and PhDStudentY

This can be explained in the following way: the concept *professor* has three properties defined in the context of publication, including *Publication*, *Published in*, and *Editor of* (see Figure 5.3). It can also be seen that the concept *professor* shares two properties (*Publication* and *Published in*) within this context with the concept *PhDStudent*.

Likewise, the similarity of the pair *(professorX, PhDStudentY)* at the instance level within the context of publication is determined as:

$$sim_{ins}(professorX, PhDStudentY : Publication)$$
$$= \frac{n_{cm}(professorX, PhDStudentY : Publication)}{n(professorX : Publication)}$$
$$= \frac{8}{21} = 0.38$$

It should be noted that the obtained values describe the similarity of a *professor* to a *PhDStudent*. Similarity of a *PhDStudent* to a *professor* would give a different value. This indicates the asymmetric feature of the approach, and is reflected by the prominence of the concept. Such a result is in the agreement with Tversky's claim [72].

In the next step, since the concepts belong to the definition and instance levels of ontology, degrees of membership of the obtained similarities are computed based on membership functions presented in Figure 5.1.a and 5.1.b, respectively. The results are shown in Table 5.2.

Table 5.2 Degrees of membership of similarity for professor and PhDStudent

$sim(A,B:R_c)$	similarity value	$\mu_{low}(u)$	$\mu_{high}(u)$
$sim_{def}(A,B:R_c)$	0.66	0.42	0.58
$sim_{ins}(A,B:R_c)$	0.38	0.65	0.35

As can be seen in Table 5.2, the computed similarity value of *professor* and *PhDStudent* at the definition level is equal to 0.66, while similarity of their instances – at the instance level – is 0.38. As it can be observed, degrees of membership to the *low* and *high* fuzzy sets of both similarities are different. If we perform a simple defuzzification in the form of an α -cut for α =0.5 for both similarities, we obtain that similarity at the definition level is *high*, while at the instance level is *low*. However, if α =0.65 then similarity is *low* at the instance level, while similarity at the definition level is undetermined.

In order to evaluate our approach, we selected one existing method related to each similarity model as described in Section 5.1. In order to make the comparison fair and meaningful the similarity value of our method is the context-free value obtained in Eq. (5.3), and without applying the fuzzy memberships. In other words, the shown result of our method is the raw calculated similarity value. The ontology built for this section, presented in Figure 5.2.a, is used in the comparison. The evaluation of results is performed using human judgment of similarity for which we averaged the similarities given by 25 graduate students for each pair of

concepts, see Table 5.3. As it can be observed, the proposed similarity assessment performs quite well comparing with other techniques as well as human judgment.

Table 5.3 Comparison of multiple similarity assessment methods

sim. model / pair	Edit distance	Shortest Path length	Leacock & Chodorow [37]	Wu & Palmer [74]	Tversky index [72]	Dice index [20]	Our method	Human judgment
professor, phd_student	0.09	0.17	0.48	0.25	0.33	0.56	0.61	0.60
professor, researcher	0.11	0.50	0.95	0.75	0.88	0.90	0.90	0.72
researcher, phd_student	0.09	0.20	0.48	0.25	0.40	0.67	0.71	0.82
professor, admin_staff	0.05	0.33	0.78	0.57	0.11	0.30	0.47	0.34

An interesting observation can be made here: the result from our proposed method is closest to the ones from Tversky and Dice index, which are also feature-based similarity models. Although the results from feature-based similarity model give a reasonably close value to the human judgment but still there is a room to be further improved. We believe that this improvement can be accomplished by taking the abstraction level of concepts in ontology into the consideration, which is performed in our approach by utilizing fuzzy set theory.

5.3 Discussion

The determination of similarity at two different levels – definition and instance – creates an opportunity to mimic human's way of similarity assessment of two items in which a person uses his/her knowledge about the categories that these two items belong to. It seems quite

plausible to assume that once the items are classified, in other words categorized, the first similarity assessment is done at the level of abstraction of concepts. Next, the person looks at the details of the items under evaluation and adjusts his/her first assessment. At the same time it is reasonable to state that the first assessment has some influence on the second assessment. In an attempt to model such a process we propose the following procedure for applying the influence of similarity obtained at each level of abstraction.

As stated, the similarity obtained at the definition level– sim_{def} – is fuzzified. Two common membership functions are used here: μ_{low} and μ_{high}. The result, i.e., two membership values $\mu_{low}(sim_{def})$ and $\mu_{high}(sim_{def})$ are combined with the instance level similarity – sim_{ins}. One of the possible ways of calculating this is presented here. The final similarity value can be obtained as:

$$similarity = \begin{cases} \max\{0, sim_{ins} - \alpha * \mu_{low}(sim_{def})\} \\ \text{when } \mu_{low}(sim_{def}) \geq \mu_{high}(sim_{def}) \\ \\ \max\{1, sim_{ins} + \beta * \mu_{high}(sim_{def})\} \\ \text{when } \mu_{low}(sim_{def}) < \mu_{high}(sim_{def}) \end{cases} \quad (5.5)$$

where α and β represent levels of influence of sim_{def} on the final similarity value. If the membership degree of sim_{def} in *low* fuzzy set is larger or equal to its membership degree in *high* fuzzy set, then the sim_{ins} is decreased by the user-defined fraction (α) of sim_{def}. In the other case, sim_{ins} is increased by the user-defined fraction (β) of sim_{def}. The final calculated similarity in Eq. (5.5) can be further translated into a human-friendly linguistic description.

Let us take the results from the example presented in the previous section. The obtained values are: 0.42 for $\mu_{low}(sim_{def})$ and 0.58 for $\mu_{high}(sim_{def})$. Therefore, the second option in Eq. (5.5) is applicable. For the value of β = 0.5 – moderate influence of similarity at the definition level – we obtain the value of similarity equal to 0.64. For β =0.75 – higher influence of definition-based similarity – the similarity value is 0.79. For β =1.0 the similarity is a summation of both similarities at the definition and instance levels (of course, if the sum exceeds 1.0 the similarity value assumes 1.0).

Chapter 6

6 Similarity assessment in Linked Data using possibility theory[36]

Linked Data (LD) represents each entity (resource) via features associated with it. In a nutshell, the proposed approach here identifies resources that are certainly shared and possibly shared between two entities, and uses elements of possibility theory to assess similarity between these entities.

6.1 Similarity measure in Linked Data

As mentioned before, LD is a mesh of interconnected resources, which can be represented as a set of triples <resource-as-subject, property, resource-as-object>. Formally:

$$LD = \{< r_i, p_q, r_m >: r_i, r_m \in R, p_q \in P\} \quad (6.1)$$

where *R* is a set of resources, and *P* is a set of properties. In this mesh, a single resource r_i is defined via its connections to other resources. Each of these resources can be considered as a feature of r_i. A set of all resources (features) connected to r_i can be treated as its semantic definition. The connections between the resource r_i and other resources are labeled with properties that have r_i as their subject. Therefore, for an entity r_i we can write:

[36] P. D. Hossein Zadeh and M. Z. Reformat. (2013) Context-aware Similarity Assessment within Semantic Space Formed in Linked Data. Journal of Ambient Intelligence and Humanized Computing. volume 4, issue 4. pp. 515-532. (published)

$$n^i = |\{< r_i, p_q, r_m >: r_m \in R \setminus \{r_i\}, p_q \in P\}| \quad (6.2)$$

where the symbol |.| stands for cardinality of a set, and n^i represents the number of connections between r_i and other resources in LD. In other words, n^i represents the number of resources – features – of r_i.

There exist four different scenarios that can be encountered during similarity assessment of two resources r_i and r_j, as shown in Table 6.1.

Table 6.1 Possible scenarios of connections between two resources

scenario label	properties (connections)		resources (features)
S1	same type		same (shared) resources
S2	same type	connecting	different resources
S3	different types		same (shared) resources
S4	different types		different resources

Let the sets P_i and P_j represent properties of the resources r_i and r_j, respectively. R_i and R_j, on the other hand, represent sets of features (connected resources) of r_i and r_j. Additionally, we define the following sets:

- a set of resources connected to both resources r_i and r_j, and a set of properties shared by both of them:

$$R_{i,j} = R_i \cap R_j \qquad P_{i,j} = P_i \cap P_j \tag{6.3}$$

- a set of resources describing exclusively the resource r_i (r_j):

$$R_i^{exc} = R_i \setminus R_j \qquad R_j^{exc} = R_j \setminus R_i \tag{6.4}$$

- likewise, a set of properties exclusive for r_i (r_j):

$$P_i^{exc} = P_i \setminus P_j \qquad P_j^{exc} = P_j \setminus P_i \tag{6.5}$$

Based on the definitions above, the scenarios for the resource r_i with respect to any resource r_j can be presented in the following way. Number of resources describing r_i that belongs to the scenario S1 is:

$$n^i S1 = \mid \{(< r_i, p_q, r_m >, < r_j, p_q, r_m >) : r_m \in R_{i,j}, p_q \in P_{i,j}\} \mid \tag{6.6}$$

scenario S2:

$$n^i S2 = \mid \{< r_i, p_q, r_m > : r_m \in R_i^{exc}, p_q \in P_{i,j}\} \mid \tag{6.7}$$

scenario S3:

$$n^i S3 = \mid \{(< r_i, p_q, r_m >, < r_j, p_s, r_m >) : r_m \in R_{i,j}, (p_q \neq p_s) \in P\} \mid \tag{6.8}$$

scenario S4:

$$n^i S4 = \mid \{< r_i, p_q, r_m > : r_m \in R_i^{exc}, p_q \in P_i^{exc}\} \mid \tag{6.9}$$

Using the descriptions given to the different scenarios, the similarity and dissimilarity between r_i and r_j can be expressed in the following way. The similarity is solely based on scenario S1 and thus the *necessity of similarity* can be determined according to the possibility theory:

$$N(sim[r_i, r_j]) = \frac{n^i S1}{n^i} \tag{6.10}$$

This represents similarity of r_i to r_j; this leads to an asymmetric nature of the proposed approach.

The *necessity of dissimilarity* of r_i to r_j is determined based on scenario S4 using the equation:

$$N(dissim[r_i, r_j]) = \frac{n^i S4}{n^i} \tag{6.11}$$

Scenarios S2 and S3 contribute to ambiguity; thus, they are involved in determining *possibility of dissimilarity*:

$$\prod(dissim[r_i, r_j]) = \frac{n^i S2 + n^i S3 + n^i S4}{n^i} \tag{6.12}$$

Therefore, similarity between resources can be expressed as an interval with $N(sim)$ as its lower limit and $\prod(sim)$ as its upper limit.

Remark. In order to remedy the computational complexity of $\prod(dissim[r_i, r_j])$ in Eq. (6.12) we may take advantage of the fact that:

$$n^i = n^i S1 + n^i S2 + n^i S3 + n^i S4 \tag{6.13}$$

Consequently, we can derive final values of *necessity of similarity* and *possibility of similarity* as:

$$N^F(sim[r_i, r_j]) = N(sim[r_i, r_j]) \tag{6.14}$$

$$\prod^F(sim[r_i, r_j]) = 1 - N(dissim[r_i, r_j]) \tag{6.15}$$

In the light of that, we can manipulate the above formulas by investigating the scenarios 2 and 3. The process for scenario S2 involves identifying similarity between these different resources, for which we have developed an algorithm. For scenario S3 the process is more complex than the one for S2. It requires an investigation of semantics of properties, which is external to LD knowledge sources. If such processes for S2 and S3 are performed the formulas for possibility of similarity and necessity of dissimilarity should be adapted accordingly as below:

$$\prod(sim[r_i, r_j]) = \frac{n^i S1 + n^i S2^C + n^i S3^C}{n^i} \tag{6.16}$$

$$N(dissim[r_i, r_j]) = \frac{n^i S4 + (n^i S2 - n^i S2^C) + (n^i S3 - n^i S3^C)}{n^i} \tag{6.17}$$

where n^iS2^C and n^iS3^C represent numbers of resources and properties indirectly related to each other.

In many real-life situations the user might be interested in similarity between two entities only in the aspect of some specific properties. This means that only those specific types of connections should be used for similarity determination. We refer to this situation as context-aware similarity assessment. We express the necessity of similarity within a context P_{cntx} as:

$$N^F_{cntx}(sim[r_i, r_j]) = \frac{n^i S1(P_{cntx})}{n^i(P_{cntx})} \tag{6.18}$$

and, the possibility of similarity as:

$$\prod^F_{cntx}(sim[r_i, r_j]) = \frac{n^i S1(P_{cntx}) + n^i S2^C(P_{cntx}) + n^i S3^C(P_{cntx})}{n^i(P_{cntx})} \tag{6.19}$$

let $n^iS2^C(P_{cntx})$ denote the number of resources in R_i (in scenario S2) that are connected indirectly to resources in R_j through some other external resources.

6.2 Experimental evaluation and comparison

We use DBPedia as the data source of RDFs of the following resources. Four movies: Matrix[37], Matrix_Reloaded[38], Hangover[39], and Blade_Runner[40], one soundtrack album: Matrix-

[37] http://dbpedia.org/page/The_Matrix
[38] http://dbpedia.org/page/The_Matrix_Reloaded
[39] http://dbpedia.org/page/The_Hangover_(film)
[40] http://dbpedia.org/page/Blade_Runner

music[41] (movie Matrix soundtrack) and one car brand: Toyota[42] are selected. First, all RDF triples associated with each resource are extracted from DBPedia. A graphical visualization of the selected resources are depicted in Figure 6.1 using a Java-based graph visualization software, Gephi[43].

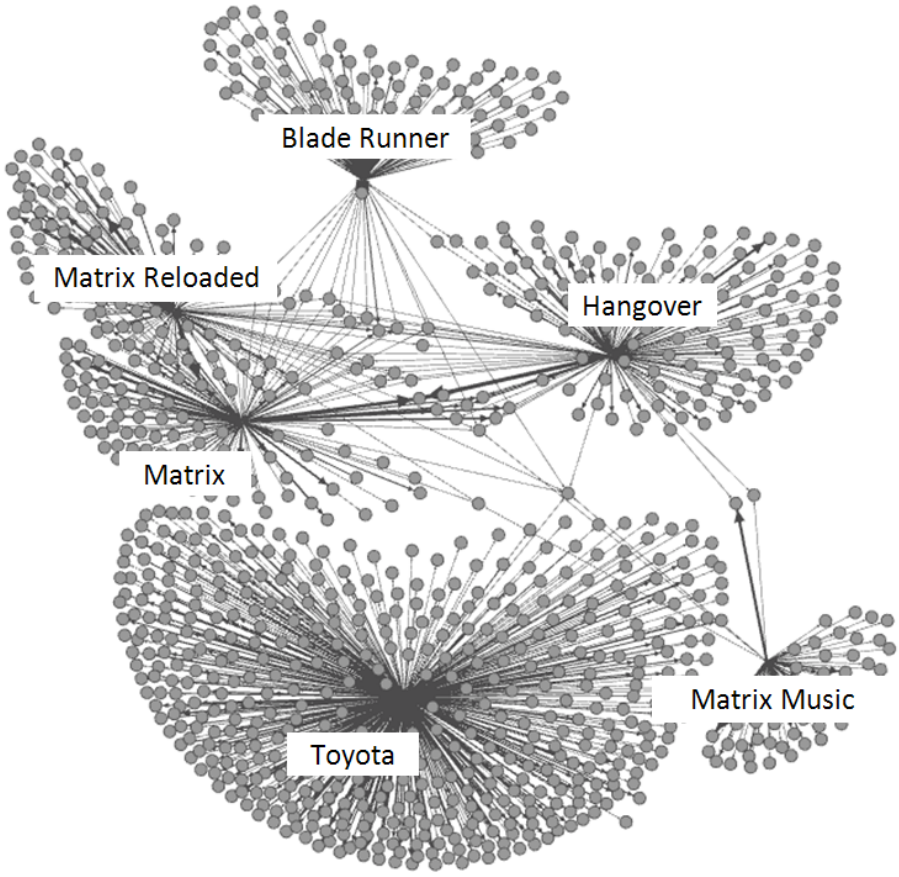


Figure 6.1 Graphical visualization of the resources described in DBPedia dataset

[41] http://dbpedia.org/page/The_Matrix:_Music_from_the_Motion_Picture
[42] http://dbpedia.org/page/Toyota
[43] http://gephi.github.io/

The evaluation is performed on the ordered pairs of resources with necessity of similarity and possibility of similarity as lower-bound and upper-bound values, as shown in Table 6.2.

Table 6.2 Context-free similarity values

Ordered pairs {ri,rj}	Necessity of similarity $N^F(sim[r_i, r_j])$	Possibility of similarity $\prod^F(sim[r_i, r_j])$	Similarity interval $sim[r_i, r_j]$
{Matrix, Matrix-Reloaded}	0.35	0.95	<0.35,0.95>
{Matrix, Blade-Runner}	0.09	0.88	<0.09.0.88>
{Matrix, Hangover}	0.09	0.94	<0.09,0.94>
{Matrix, Matrix-music}	0.01	0.73	<0.01,0.73>
{Matrix, Toyota}	0.00	0.71	<0.00,0.71>

In general, there are two important observations that can be made at this point. Firstly, the necessity of similarity gives an unquestionable similarity value, and thus it is referred to as a lower-bound of similarity. The possibility of similarity, on the other hand, is determined based on the necessity of dissimilarity. Secondly, the interval is a range of possible values of similarity between resources. Overall, it can be inferred that context-free similarity provides an unbiased measure of similarity between two resources based on all the available information about the resources without taking into account any consideration.

Table 6.3 Context-aware similarity values

Ordered pairs {ri,rj}	Context	Necessity of similarity $N^{F}_{cntx}(sim[r_i, r_j])$	Possibility of similarity $\prod^{F}_{cntx}(sim[r_i, r_j])$	Similarity interval $sim[r_i, r_j]$
{Matrix, Matrix-Reloaded}	starring	0.80	0.85	<0.80,0.85>
{Matrix, Blade-Runner}	subject	0.37	0.48	<0.37,0.48>
{Matrix, Hangover}	distributor	1.00	1.00	<1.00,1.00>
{Matrix, Matrix-music}	Type	0.08	0.19	<0.08,0.19>
{Matrix, Toyota}	label	0.00	0.00	<0.00,0.00>

Results of context-aware similarity between the same set of pairs are shown in Table 6.3. As it can be seen, the uncertainty intervals in context-aware similarities are narrower than in context-free measures, which means higher confidence in context-aware similarity measures. This type of similarity is more often used in real-life scenarios, especially in situations involving human judgment.

Numbers of feature-based methods are selected and compared to our method. We have selected 12 pairs of real-world entities extracted from DBpedia as shown in Table 6.4. The pairs from #1 to #3, #4 to #8, and #9 to #12 are selected as very similar, relatively similar, and dissimilar entities, respectively.

Table 6.4 Comparison of our approach to other related methods

		Similarity models				
		Corpus-based	Feature-based		Concept-based	LD-based
#	Ordered pairs	Latent Semantic Analysis [36]	Tversky [72]	Dice [20]	Boros [8]	Our method
1	{Matrix, Matrix-reloaded}	0.96	0.40	0.38	0.35	<0.35,0.95>
2	{Good-fellas, God-father}	0.92	0.29	0.37	0.20	<0.20,0.97>
3	{Jaws, Jurassic-Park}	0.87	0.54	0.68	0.20	<0.20,0.90>
4	{Matrix, Matrix-music}	0.90	0.25	0.20	0.01	<0.01,0.73>
5	{Star-wars, Star-trek}	0.79	0.42	0.36	0.30	<0.30,0.56>
6	{Jurassic-Park, Godzilla}	0.66	0.02	0.03	0.02	<0.02,0.24>
7	{Spider-man, I-robot}	0.75	0.25	0.30	0.10	<0.10,0.55>
8	{Matrix, Blade-runner}	0.85	0.55	0.4	0.09	<0.09,0.88>
9	{Matrix, Hangover}	0.87	0.10	0.15	0.09	<0.09,0.94>
10	{Matrix, Toyota}	0.58	0.20	0.32	0.00	<0.00,0.71>
11	{Pulp-fiction, Wall-E}	0.89	0.09	0.09	0.08	<0.08,0.15>
12	{Angry-birds, Titanic}	0.63	0.04	0.05	0.02	<0.02,0.10>

The second set of experiments is an attempt to compare our proposed method with taxonomy-based measures. For this reason, we use DBpedia ontology[44], which consists of more than 320 classes, where these classes are organized in the hierarchy with maximum depth of 7. The obtained results in Table 6.5 suggest the inadequacy of the selected taxonomy-based measures for the pairs #1 to #3, #5 to #9, and #11.

Table 6.5 Similarity values of taxonomy-based methods (pair# is taken from Table 6.4)

Pair#	Wu&Palmer [74]	Leacock & Chodorow [37]	Our method
#1, #2, #3, #5, #6, #7, #8, #9, #11	0.67	1.1	<0.35,0.95>,<0.20,0.97>,<0.20,0.90>,<0.30,0.56>, <0.02,0.24>,<0.10,0.55>,<0.09,0.88>,<0.09,0.94>, <0.08,0.15>
#4	0.34	0.48	<0.01,0.73>
#10	0.30	0.42	<0.00,0.71>
#12	0.34	0.48	<0.02,0.10>

[44] http://wiki.dbpedia.org/Ontology

Chapter 7

7 Fuzzy semantic similarity in Linked Data using the OWA operator[45]

In the following approach, we provide a novel solution for determining similarity between concepts in LD while allowing the importance of properties to influence the obtained similarity measure. Our proposed approach is implemented based on feature-based similarity model, which considers the shared objects between the concepts. First, we develop a membership function to capture the importance of different properties, and then we use ordered weighting averaging (OWA) operator for aggregation of multiple similarity measures corresponding to different importance levels of properties.

7.1 Fuzzy similarity of concepts based on importance of properties

The LD can be represented as a set of triples:

$$LD = \{< c_m, p_z, d_n > \mid \ c_m \in C, d_n \in C \cup D, p \in P\} \tag{7.1}$$

[45] P. D. Hossein Zadeh and M. Z. Reformat. (2012) Fuzzy Semantic Similarity in Linked Data using the OWA Operator. 2012 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS). pp: 1-6. (published)

where *C*, *D*, and $P=\{p_1,p_2,\ldots,p_u\}$ are sets of concepts, data properties and object properties, respectively. Each concept c has a number of features, d, connected to it via properties, p. Therefore, we can represent a concept with its associated describing features.

Without losing any generality, we can assume that the set of properties can be classified into a number of subsets with each subset containing properties that are equally important. In other words, we define a set *L* of *n* subsets describing all importance levels of properties in a concept (in a descending order from most important to least important sets of properties):

$$L = \{l_1, l_2, \ldots, l_n\} \tag{7.2}$$

Each property, *p*, belongs to only one subset in *L* according to its semantic influence in that concept. However, each subset l_i may contain multiple properties. Thus, similarity (called hereafter *propertySimilarity*) between the two concepts *X* and *Y* is determined for each subset, l_i, as follows:

$$S_i(X,Y) = \frac{n_i(X,Y)}{n_i(X)} \tag{7.3}$$

where $n_i(X,Y)$ is the number of common features that reside in the subset l_i between the two concepts *X* and *Y*. Also, $n_i(X)$ is the total number of features associated with the subset l_i connected to the concept *X*.

The final similarity, S^f, is the aggregation of *propertySimilarity* values in Eq. (7.3) and is expressed as:

$$S^f = aggr(S_1, S_2, \ldots, S_n) \tag{7.4}$$

We define membership functions to reflect the degree of contribution of each *propertySimilarity*. Membership degree for each *propertySimilarity* has the form:

$$\mu(S_i) = (S_i)^{\psi_i} \tag{7.5}$$

where ψ is introduced as a significance factor and is obtained as:

$$\psi_i = i - f(l_i, c) \tag{7.6}$$

i – an index representing importance level of properties.

$f(l_i, c)$ – a ratio of a number of properties of a subset l_i for a given concept to the total number of properties over all l_is of that particular concept.

We use importance weighted quantifier guided (OWA) aggregation to combine the *propertySimilarity* values. An OWA operator *F* is a mapping $F : \Re^n \to \Re$ and is given by [75]:

$$aggr(S_1, S_2, \ldots, S_n) = \sum_{j=1}^{n} b_j . w_j(\mathbf{m}) \tag{7.7}$$

where b_j is the j^{th} largest value in $\{S_1, S_2, \ldots, S_n\}$ while S_is are in descending order, and $\mathbf{m} = [\mu(b_1), \mu(b_2), \ldots, \mu(b_n)]$ is a vector containing the membership degrees of *propertySimilarity* values calculated using Eq. (7.5).

The weight corresponding to the j^{th} element in $\{S_1, S_2, \ldots, S_n\}$ is given by [75]:

$$w_j(\mathbf{m}) = Q\left(\frac{\sum_{k=1}^{j} m_k}{T}\right) - Q\left(\frac{\sum_{k=1}^{j-1} m_k}{T}\right) \qquad 1 \le j \le n \tag{7.8}$$

where $T = \sum_{k=1}^{n} m_k$, and Q is of the form $Q(r) = r^{0.5}$, and $m_k \in [0,1]$ denotes the membership degree associated with the k^{th} largest value in $\{S_1, S_2, \ldots, S_n\}$. Finally, overall similarity in Eq. (7.4) is calculated using Eq. (7.7) and Eq. (7.8).

Remark. Assume all properties of a concept are equally important in the process of similarity assessment. This resembles a generalized feature-matching similarity method for two concepts X and Y as presented by Tversky [72], Dice [20], etc.:

$$S(X,Y) = \frac{n(X,Y)}{n(X)} \tag{7.9}$$

We shall denote this as a special case of our approach in Eq. (7.3).

7.2 Experimental study

For experiment, numbers of instances of a concept "book" from a real-world dataset DBpedia are extracted. SPARQL query language is used in order to query the RDF triples of the instances. We define four subsets of properties that are put together according to their importance in the measure of similarity of a concept "book", as shown in Table 7.1.

Table 7.1 Four subsets for properties of the concept "book"

l_1	{dbpedia-owl: author, dbpedia-owl: literaryGenre, dbprop: author, dbprop: genre, dbprop: name, dbprop: title, dbprop: englishTitle, dcterms: subject, foaf: name}
l_2	{dbpedia-owl: language, dbprop: type, dbprop: language, rdf: type, foaf: primaryTopic of}
l_3	{dbpedia-owl: series, dbpedia-owl: subsequentWork, dbpedia-owl: noteAbleWork of, dbpedia-owl: previousWork of, dbpedia-owl: publisher, dbprop: series, dbprop: followedBy, dbprop: precededBy of, dbprop: releaseDate, dbprop: publisher, dbprop: country}

l_4	{dbpedia-owl: numberOfPages, dbpedia-owl: coverArtist, dbpedia-owl: mediaType, dbprop: pages, dbprop: lang, dbprop: langtitle, dbprop:mediaType, dbprop: titleOrig, dbprop:coverArtist, foaf:page }
null	{dbpedia-owl: abstract, dbpedia-owl: isbn, dbpedia-owl: oclc, dbpedia-owl: dcc, dbpedia-owl:lcc, dbpedia-owl: thumbnail, dbpedia-owl: wikiPageExternalLink, dbpedia-owl: wikiPageRedirects, dbpedia-owl: wikiPageDisambiguates of, dbprop: id, dbprop: isbn, dbprop: oclc, dbprop: congress, dbprop: en, dbprop: entxt, dbprop: imageCaption, dbprop: wikiPageUsesTemplate, foaf: depiction, rdfs: comment, rdfs: label}

We selected pairs of books' instances in order to calculate and compare the similarity values. The comparison is between the similarity values obtained with our method in Eq. (7.3), the non-weighted measure in Eq. (7.9), and the Tversky measure [72], for results see Table 7.2. To briefly explain the selected instances: The book "*Sicilian*" is a novel by Mario Puzo, and it is known as the sequel of the novel "*The Godfather*" written by the same author. The book "*Do Androids Dream*" is a 1968 science-fiction novel by Philip K. Dick, which is set in an earth's end of civilization era and is similar to the book *"Ubik"* in areas of science-fiction novels, existential novels, released in 1960s, Philip K. Dick as the author, same publisher, and so on. *The "Master and Margarita*" is a fantasy comedy novel, and "*Hyperion*" is a science-fiction novel by American writer Dan Simmons.

Table 7.2 Experimental results of similarity values

Pairs of Concepts	Our method S^f	Our method Non-weighted Eq. (7.9)	Tversky's method [72]
(The Godfather, Sicilian)	0.74	0.22	0.4

(Do Androids Dream, Hyperion)	0.36	0.09	0.13
(Do Androids Dream, The Godfather)	0.42	0.09	0.12
(Do Androids Dream, Ubik)	0.55	0.2	0.2
(Do Androids Dream, The Master and Margarita)	0.38	0.07	0.09

Let us discuss the results obtained by our method for the pair (The Godfather, Sicilian). As can be seen in Table 7.2, the similarity of the pair (The Godfather, Sicilian) is very high as these two concepts share quite a number of features such as author, subject, type, country, and series. Especially the majority of the shared features reside in subset I_1, which is the subset containing the most important properties to the concept “book”.

Comparing our results with [72], confirms the influence of properties’ importance levels on the similarity measure. Also, similarities obtained from the non-weighted approach are very much compatible with Tversky’s method [72] since none of these approaches consider properties classification according to their importance.

Table 7.3 Asymmetric similarity

Pairs of Concepts	Our method S^f
(Sicilian, The Godfather)	0.64
(Hyperion, Do Androids Dream)	0.41

(Ubik, Do Androids Dream)	0.65
(The Master and Margarita, Do Androids Dream)	0.41

Table 7.3 shows the asymmetric similarity in our method in a number of pairs selected from Table 7.2. Comparing to the similarity values in Table 7.2, it can be seen that similarity measures differ slightly as the order of pairs changes.

Chapter 8

8 A fuzzy semantic similarity in Linked Data using Wikipedia Infobox[46] [47]

The problem of semantic similarity assessment arises in several applications, for example, knowledge management, information integration, and information discovery. In this chapter, we present a new method that evaluates similarity between entities represented by RDF triples introduced in the context of the Semantic Web. At the beginning, our approach identifies and groups properties according to their importance. It is done via exploiting the information presented in Wikipedia infoboxes. Then semantic similarity corresponding to each group is calculated using both the schema (ontology classes and properties) and RDF links discovered from different datasets (due to the open and distributed nature of data). Finally, the calculated similarity measures for all groups are aggregated using weights obtained from a specially designed membership function. Experimental evaluations confirm the suitability of the proposed method.

This chapter introduces a novel approach suitable for identification of related items. A number of important aspects of the proposed approach are presented here:

[46] P. D. Hossein Zadeh and M. Z. Reformat. (2013) Fuzzy Semantic Similarity in Linked Data using Wikipedia Infobox. IEEE IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS) Joint. pp: 395-400. (published)

[47] P. D. Hossein Zadeh and M. Z. Reformat (2015) The Web, Similarity, and Fuzziness. 50 Years of Fuzzy Logic and its Applications, Springer. volume 326, pp: 519 - 536 (published)

- The evaluation of relatedness of two items is performed using features of the items. The RDF representation allows us to compare items on a feature-by-feature (triple-by-triple) basis. The principles of the approach are explained in Section 8.2.

- The importance of features is recognized as essential characteristics of similarity evaluation process. Different features contribute to the overall similarity in different ways. It is important to automatically determine importance of features, as well as to apply a proper aggregation process to combine similarities of individual features. The process of determining importance of features is based on Wikipedia Infoboxes[48] as explained in Section 8.3. Further, a fuzzy-based method of aggregating evaluated similarities of single features and taking into account different importance levels of the features are fully explained in Section 8.4.

- The proposed method is applied to a real-life scenario of finding relevant books. The results obtained using the proposed approach are compared with the suggestions provided by Google search engine. The case study is presented in Section 8.5.

8.1 Similarity evaluations

The principle idea presented here relies on the assertion that properties of an entity should have different importance values in similarity assessment between that entity and other entities. These importance values reflect their influence in describing that entity. Thus, similarity between entities cannot be ideally calculated with properties having equal weights. In fact, human judgment of similarity considers relative importance values for properties of an item. As an example, in a problem of finding books similar to a particular book, properties such

[48] http://en.wikipedia.org/wiki/Help:Infobox

as “author”, “genre”, and “subject” are more dominant compared to such properties as “country”, “number of pages”, and “cover artist”. It worth noting that the present study should not be confused with similarity assessment within a context defined via specific properties. For example, a context similarity evaluation can be applied in the question, “How much these two books are similar in the context of their *topic*?” For similarity assessment in a context in LD, see Chapters 6 and 7. The solution presented in this chapter determines the semantic similarity between entities expressed in RDF triples while recognizing and dealing with the importance of each property associated with the entities under study.

A fundamental step in similarity assessment of two entities is comparing features associated with the entities. Having RDF triples for representing information in LD, features are represented with properties and their values with objects (Section 8.3). For example, in LD a book can be represented with multiple features (properties) such as name, author, country, language, genre, and publisher. An example of a real entity (from DBpedia) described with multiple features is shown in Figure 8.1. The problem of handling importance of features is composed of two sub-problems: 1) How to recognize dominant properties? 2) How to deal with them?. In real life, it is intuitive to distinguish very important properties of an entity from less important and not important ones. A computer program requires a well-defined approach to do the same task since every piece of information in LD is represented as RDF triples, and all the triples are equally important. Additionally, importance values of properties for all entities are constantly modified on the web. Therefore, a process of determining the importance values is a very time-consuming and impractical task.

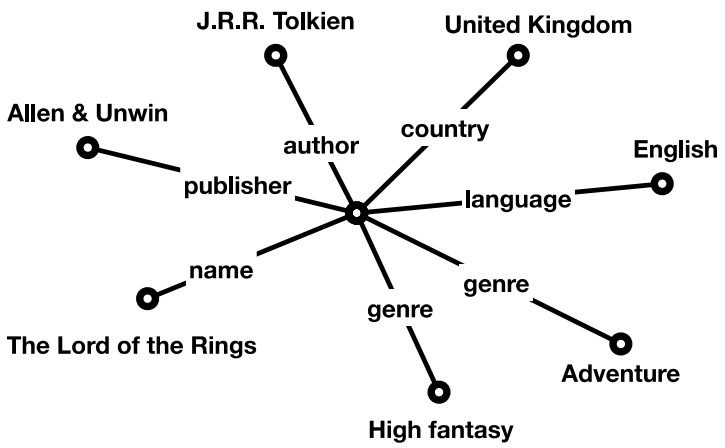


Figure 8.1 Book “The Lord of the Rings” with its features

Section 8.3 presents an answer to the first sub-problem of how to detect and distinguish the properties that vary in their importance. We present a solution to this problem by obtaining this information from the most popular online encyclopedia, Wikipedia. The next important question is how to use the obtained properties and their significance values in similarity evaluation. The answer to this question, membership functions, is explained in Section 8.4.

8.2 Layers in linked triples

The underlying idea of the proposed approach in this chapter for relevancy evaluation is to determine number of common and relevant features. In the case of RDF defined concepts, this nicely converts into checking how many features they share as presented in Figure 8.2. Defined entities are books “The Godfather” and “The Sicilian” that share number of features. Some of these features are identical – the same property and the same subject (black circles in Figure 8.2), while some have the same object but different properties.

Basically, number of different comparison scenarios can be identified. It depends on interpretation of the term “entities that they share”. The possible scenarios are (for details see Chapter 6):

- identical properties and identical objects
- identical properties and similar objects
- similar properties and identical objects
- similar properties and similar objects

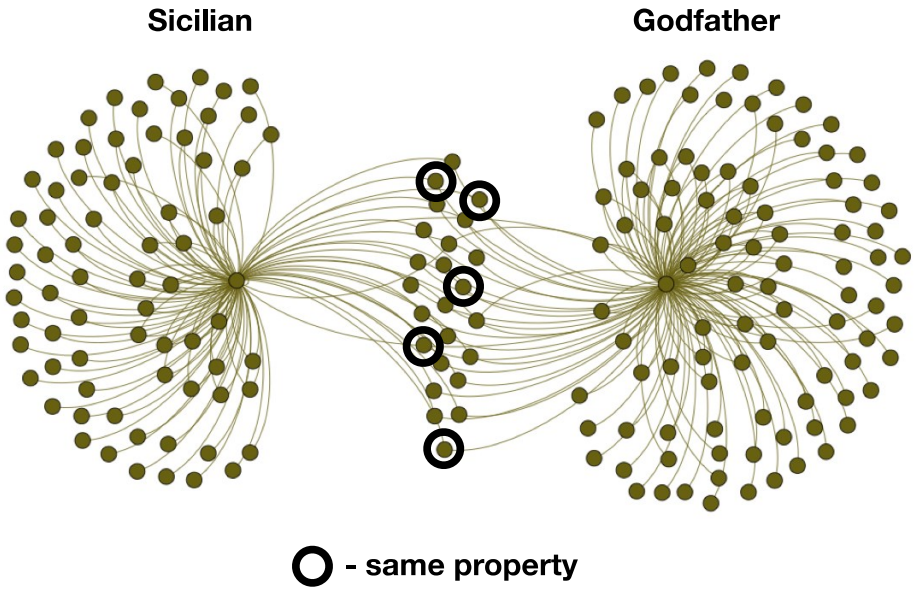


Figure 8.2 Similarity of RDF defined concepts: based on shared objects connected to the defined entities with the same properties

To better understand the proposed methodology, Figure 8.3 shows similarity assessment between two entities *x* and *y*. As can be seen, similarity is evaluated by taking into account two layers, Layer 1 and Layer 2. Similarity of Layer 1 is assessed via two components: common objects of the two entities, i.e., the common object {*z*} in Figure 8.3; and the pair of unique

objects {(*w*, *u*)}. Similarity of Layer 2 is to evaluate similarity between unique pairs – {(*w*, *u*)}. It is evaluated based on all permutations of objects that are connected to the elements, i.e., {(*s*, *g*), (*s*, *f*), (*s*, *r*), (*t*, *g*), (*t*, *f*), and (*t*, *r*)}.

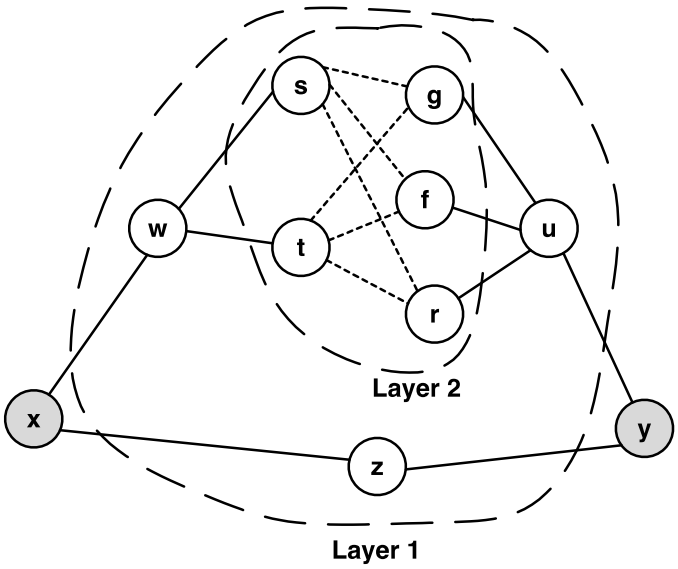


Figure 8.3 Similarity evaluation process for Layer 1 and Layer 2

8.3 Properties and their importance

Once the properties and their importance values are defined, we create a fuzzy set *L* of *n* subsets that categorize the properties with respect to their importance. Each subset has an assigned linguistic label that corresponds to its importance degree, such as:

$$L = \{l_1 = critical, l_2 = very\ important, \ldots, l_n = not\ important\} \tag{8.1}$$

In other words, properties with equal importance describing an entity are classified in the same subset. Each subset l_i may contain any number of properties:

$$l_i = \{p_1, p_2, \ldots, p_m\} \tag{8.2}$$

A total number of subsets, *n*, is a user-defined constant. This helps us to categorize a property in a proper subset that indicates its importance.

We developed an approach using Wikipedia Infoboxes to find key properties of an entity and to classify them into the suitable importance subset. Infobox is a summarized information represented in a table on the top right-hand side of a Wikipedia page. It provides information about a particular entity. An example of Infobox for the book “The Lord of the Rings” is shown in Figure 8.4.

Figure 8.4 Wikipedia Infobox for the book “The Lord of the Rings”

First, we obtain the Infobox template[49] corresponding to the category of a considered entity. Properties included in the Infobox template are selected as the characteristic properties of the entity that we keep. We discard the rest of properties that the entity has. This step reduces the amount of data to be processed in a similarity evaluation method. Next, we classify the properties into proper subsets of *L* based on their importance in describing the entity.

The main idea proposed here is to exploit the information in the *domain* of a property. Domain of a property is a class of the subject in the <subject-property-object> RDF triple. Basically, the class refers to an item located in a hierarchical taxonomy. We argue that this information plays a critical role in identifying the importance of a property. Therefore, we categorize the properties based on the location of their domains in taxonomy of domains. This approach is justified because classes located in higher levels of taxonomy are more abstract than the ones in lower levels. In general, abstract classes carry paramount description of an entity compared to less abstract and specific classes. Thus, properties with more abstract domains are more important. For example, considering an entity "book" properties such as {subject, genre, name} carry important information, intuitively, and they belong to the most abstract class, "thing", in DBpedia ontology[50]. In Figure 8.5, a fragment of DBpedia taxonomy related to an entity "book" is depicted.

[49] http://en.wikipedia.org/wiki/Template:Infobox_book
[50] http://mappings.dbpedia.org/server/ontology/classes/

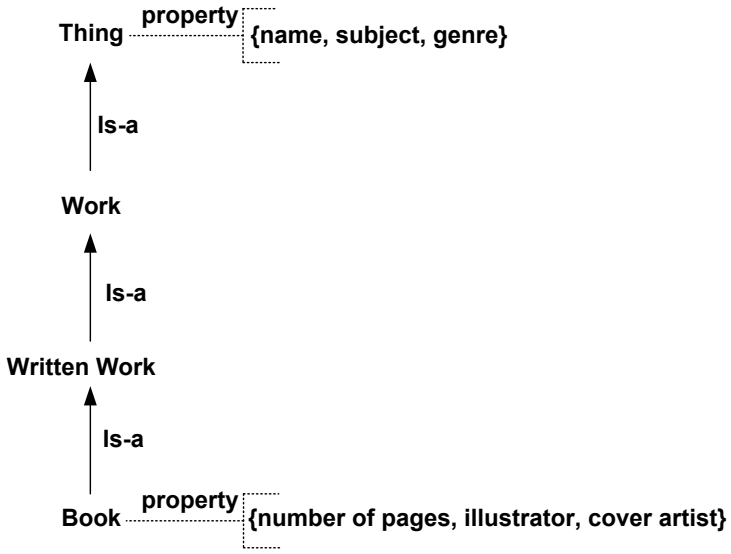


Figure 8.5 Small fragment of DBpedia taxonomy for an entity "book"

In a situation of comparing entities that belong to different Infobox templates, e.g., a book and a car, same process is followed. However, the obtained similarity will be very low as it lacks existence of common properties.

In LD, information is represented as a set of triples:

$$LD = \{< s, p, o > \mid \ s \in C, p \in P, o \in C \cup D\} \tag{8.3}$$

where C, D and $P=\{p_1,p_2,...,p_m\}$ are sets of entities, data values and properties, respectively. Each entity, c, is a subject in a number of triples connected to it via properties, p. Therefore, we represent an entity as a set of triples defining it.

Our proposed algorithm for similarity calculation is shown in details in pseudo codes in Table 8.1 and Table 8.2. In line 5 (Table 8.1), a set of common triples between two entities x and y, O_{common}, is obtained. O_x and O_y are two sets of objects unique to each entity x and y, respectively. They are initialized in line 6. For all permutations of elements in sets O_x and O_y, a

sub-function *Sim_Second_Layer* is called (line 10). Similarites calculated for all pairs (*a*, *b*) are obtained and combined in line 12. Similarity between two entities *x* and *y* is calculated in line 16. It should be noted that to avoid division by zero, which happens in the case if there are no common objects, the value of similarity in line 8 is set to zero. This situation may happen when different entities are compared. An average over similarities related to all properties (initialized in line 3) leads to the similarity of x and y at the level l_i (line 14).

Table 8.1 Pseudo code for similarity calculation

Similarity_final(x,y)

1 $L = \{l_1, l_2, ..., l_n\}$ */* initializing set L */*

2 for $\forall l_i \in L$

3 $l_i = \{p_1, p_2, ..., p_m\}$ */* initializing set* l_i **/*

4 for $\forall p_j \in l_i$

5 O_{common} */* initialize set of objects common to x and y attached via property* p_j **/*

6 O_x, O_y */*initialize set of objects unique to x and y attached via property* p_j **/*

7 *sim_layer1=|* O_{common} *|*

8 *sim_layer2=0*

9 *for* $\forall(a,b) \mid a \in O_x \;\&\; b \in O_y$

10 *sim_layer2=sim_layer2+**Sim_Second_Layer(a,b)***

11 *end*

12 $Sim^{l_i}_{p_j} = \frac{sim\_layer1 + sim\_layer2}{|\,O_{common}\,| + |\,O_x\,| \,.\, |\,O_y\,|}$

13 *end*

14 $Sim^{l_i} = avg\left(Sim^{l_i}_{p_1}, Sim^{l_i}_{p_2}, \ldots, Sim^{l_i}_{p_m}\right)$

15 *end*

16 $Sim^{final} = aggr\left(Sim^{l_1}, Sim^{l_2}, \ldots, Sim^{l_n}\right)$

In sub-function *Sim_Second_Layer*, triples of the pair of entities *a* and *b* are extracted, Table 8.2. Two sets of O_a and O_b are initialized each containing the objects attached to entities *a* and *b* respectively via the property “rdf:type” (line 1). Note that, only triples having the property “rdf:type” are obtained and the rest are discarded. This is because, the description of an entity provided by the property “rdf:type" is of a special importance in LD. “rdf:type” is used to say that things are of certain types. It is worth noting that a similar procedure can be

repeated for the property “rdf:subject”. Results from these two properties may be combined depending on how the information is expressed in a data set. For simplicity, we only consider the property “rdf:type” in the proposed similarity computation process. Similarity between *c* and *d* as the permutations of elements in sets O_a and O_b is calculated in lines 5-8. If *c* and *d* are different, their similarity is calculated using (Wu and Palmer 1994). Otherwise, their similarity is calculated based on the depth of the ontology that *c* or *d* belongs to. Finally, the maximum of similarities of all pairs is returned to the main function *Similarity_final(x,y)* (line 11).

Table 8.2 *Pseudo code of Sim_Second_Layer* (a, b)

Sim_Second_Layer(a,b)

1 O_a, O_b */* initializing sets of objects attached to a and b via property rdf:type */*

2 $k = 0$

3 ***for all pairs*** (*c, d*) ***such that*** $c \in O_a, d \in O_b$

4 $k = k + 1$

5 ***if*** $c \neq d$

6 $Sim_{[k]}(c,d) = \frac{2 * depth(LCS(c,d)}{depth(c) + depth(d)}$ */* LCS: least common subsume of c and d in ontology */*

7 ***else if***

8 $$Sim_{[k]}(\mathrm{c},\mathrm{d}) = 1 - \frac{1}{depth(\mathrm{c})}$$

9 ***end***

10 ***end***

11 ***return*** *max_over_k(* $Sim_{[k]}(\mathrm{c},\mathrm{d})$ *)*

8.4 Similarity and fuzziness

Similarity between two entities x and y is defined as the aggregated similarity values computed for every subset l_i:

$$Sim^{final}(x,y) = aggr(Sim^{l_1}(x,y), Sim^{l2}(x,y), \ldots, Sim^{l_n}(x,y)) \quad (8.4)$$

where *aggr(.)* is an aggregation operator, described later. Similarity values related to each subset l_i is obtained as the average of similarities for all properties in that subset:

$$Sim^{l_i}(x,y) = avg(Sim^{l_1}_{p_1}(x,y), Sim^{l_1}_{p_2}(x,y), \ldots, Sim^{l_1}_{p_m}(x,y)) \quad (8.5)$$

The *Sim(.)* function calculates the similarity value between two entities as defined below. Due to the nature of LD entities along with their properties are distributed over the Web in a form of connected RDF triples. Considering an entity in LD, values of its properties may be a

subject of another triple and so on. Those triples provide further information that can be used in similarity evaluation of an entity to another. For this reason, we include in similarity evaluation not only the triples that are directly connected to the entity (Layer 1) but also the ones connected one layer further away from the entity (triples describing the objects of that entity), see Figure 8.3.

In Eq. (8.4), the *aggr(.)* operation can be any process that takes the weights of the similarity values into consideration. The main idea is that similarity measures calculated for each subset l_i contribute differently to the final similarity according to their importance. Here, it is defined as the normalized weighted sum of the similarity measures in which weights are the membership degrees obtained in Eq. (8.7). The final similarity is calculated as:

$$Sim^{final}(x,y) = w_1.Sim^{l_1}(x,y) + w_2.Sim^{l_2}(x,y) + \ldots + w_n.Sim^{l_n}(x,y) \tag{8.6}$$

where,

$$w_i = \frac{\mu(Sim^{l_i}(x,y))}{\sum_i \mu(Sim^{l_i}(x,y))} \tag{8.7}$$

$\mu(.)$ gives the membership degree for each $Sim^{l_i}(x,y)$ and is obtained as follows:

$$\mu(Sim^{l_i}(x,y)) = \left(Sim^{l_i}(x,y)\right)^{\psi_i} \tag{8.8}$$

Here, we know that $Sim^{l_i}(x,y) \in [0,1]$, therefore larger values of ψ_i leads to smaller values of membership degrees. ψ is a significance power and is calculated as:

$$\psi_i = i - f(l_i, c) \tag{8.9}$$

where *i* is an index of the subset l_i representing importance of a property, and $f(l_i, c)$ is a ratio of a number of properties of a subset l_i for a given entity to the total number of properties over all l_i 's of that particular entity. To justify Eq. (8.9) it should be noted that more important properties have smaller values of *i* and their $\mu(Sim^{l_i}(x,y))$ are larger. Also, $f(l_i, c)$ adjusts $\mu(Sim^{l_i}(x,y))$ such that if the subset l_i constitutes a substantial part of all properties, $f(l_i, c)$ is larger, $i - f(l_i, c)$ becomes smaller, and $\mu(Sim^{l_i}(x,y))$ increases. This shows higher influence of more representative properties.

All the steps in our approach of calculating similarity between any two entities are shown in Figure 8.6. As can be seen, triples in Layer 1 and Layer 2 of entities under similarity assessment are extracted from datasets in LD. After detecting the category of the entity by examining the related property, the corresponding Infobox template from Wikipedia is obtained. The Infobox contains key properties of that entity. We categorize them and rank the subsets of properties $L = \{l_1, l_2, \ldots, l_n\}$. Similarity values for all subsets are calculated separately, and further they are aggregated to compute the final similarity.

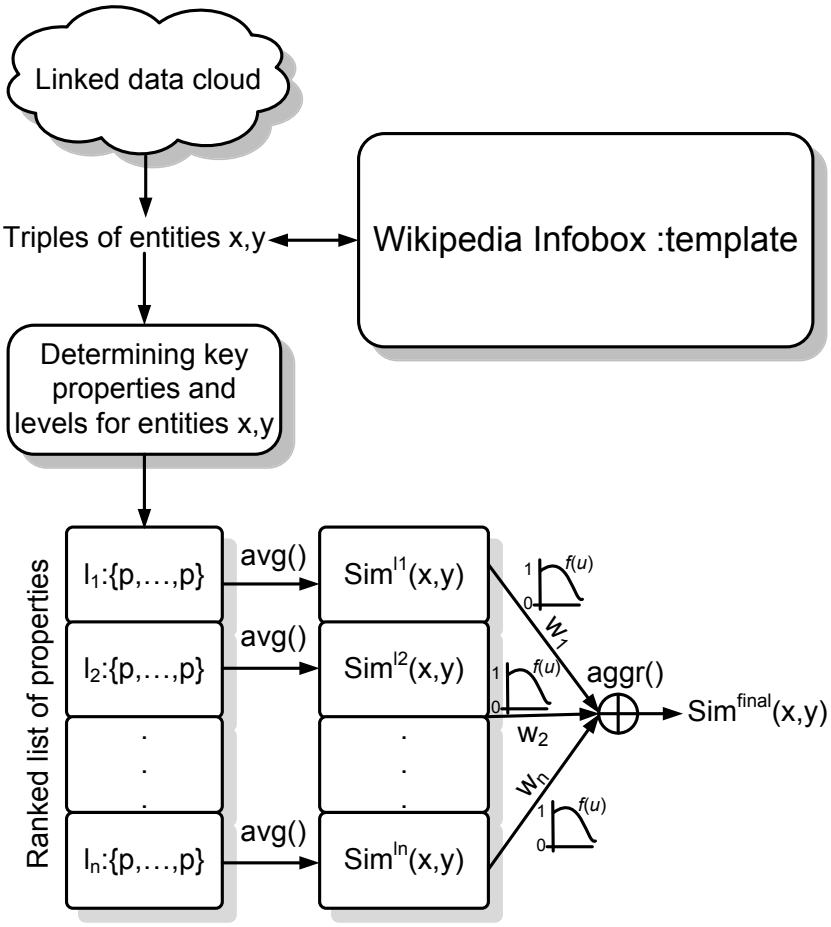


Figure 8.6 Schematic of the similarity evaluation approach

8.5 Experiments

To evaluate the above approach, a set of entities is selected from a real-world dataset DBpedia and their associated RDF triples were extracted. The entities are instances of a concept "book" in DBpedia. As discussed previously, the Infobox template representing the concept "book" is extracted. This helps to detect characteristic properties and to group them. Next, a list of classes, {book, thing, work, written work}, are obtained from this template representing domains of the Infobox properties. Accordingly, we define four subsets of properties that group

the properties within the same domain. Based on the location of each domain in the DBpedia ontology, we assign importance ranking to the created subsets. Table 8.3 shows the formed subsets for the entity "book" for this experiment.

Table 8.3 Four subsets for properties of the concept "book"

l_1	name, caption, title, country, language, series, subject, genre, publication date
l_2	author, translator, publisher, preceded by, followed by
l_3	oclc[51], lcc[52]
l_4	illustrator, cover artist, media type, number of pages, isbn, dewey[53]
null	first publication date, last publication date, number of volumes, based on, completion date, license, description, abstract, rights, editor, format, sales, etc

The last subset in Table 8.3, null, contains properties related to an entity "book" that are labeled as non-important and are ignored in the similarity evaluation process. It is worth noting that discarding the non-important properties may cause losing some information. However, this will reduce the time and increase the speed of the similarity evaluation process especially when large number of entities are described with large numbers of properties. In addition, selection of these subsets and assigning each property to a subset can be customized to users' preferences or an application context.

[51] Online Computer Library Center number
[52] Library of Congress Classification
[53] Dewey Decimal System Classification

Eight instances of the entity “book” are selected: “The Godfather”, “The Sicilian”, “Do Androids Dream of Electric Sheep?”, “Hyperion”, “Ubik”, “The Master and Margarita”, “Fools Die” and “The Family Corleone”. Figure 8.7 illustrates entities and their features, as well as relationships between. As it can be seen, entities may be connected directly via common objects or through subsequent connections.

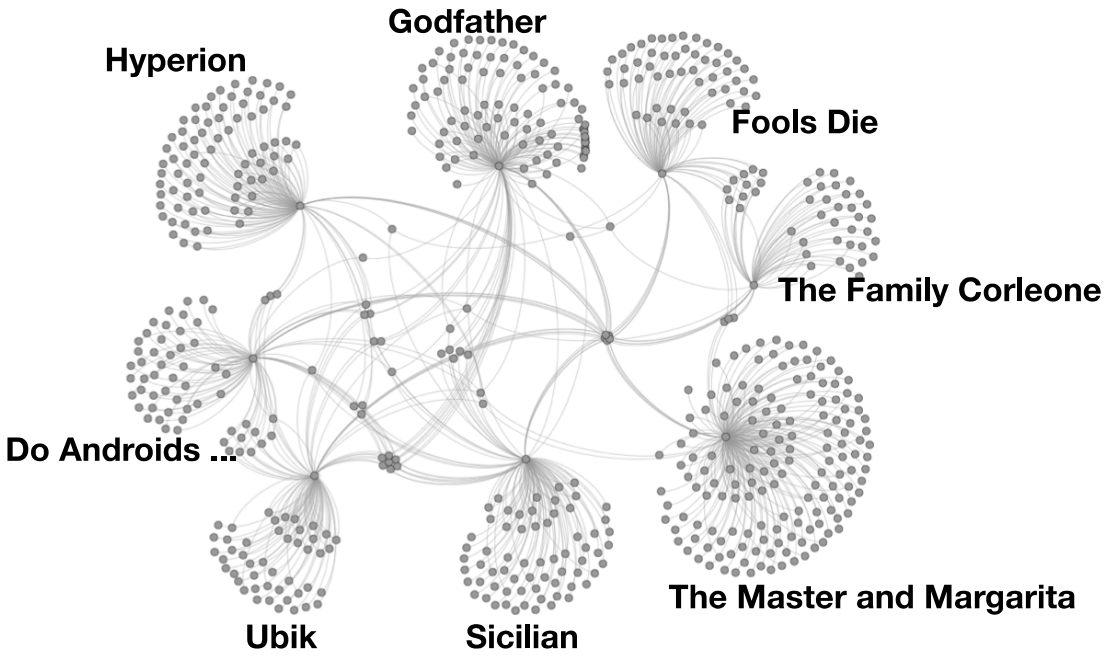


Figure 8.7 Relationship of the given entities in LD

Figure 8.8 shows similarity results between the entities based on the approach presented in Section 8.4.

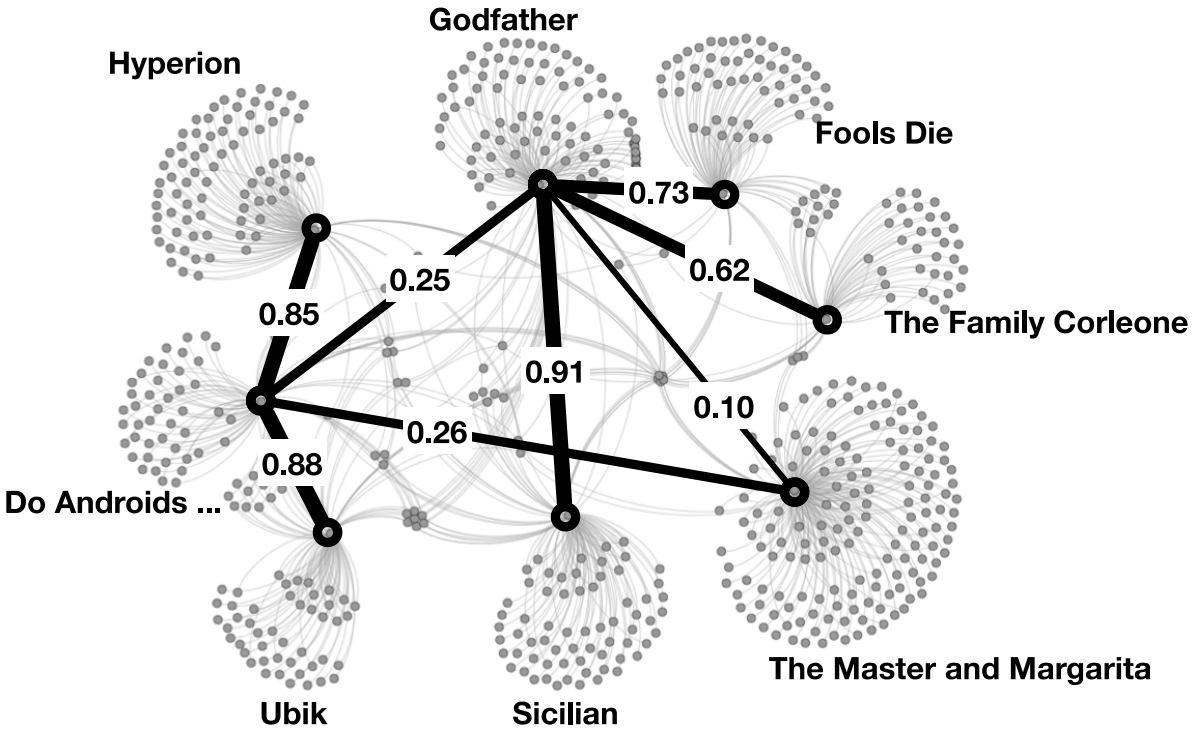


Figure 8.8 Similarity values between entities

According to the obtained values of similarity between any two pair of books, the books "The Sicilian", "The Family Corleone" and "Fools die" are ranked as top three matches. Table 8.4, compares this result to the Google Knowledge Graph and the suggestions provided by Amazon[54]. For the Google Knowledge Graph[55], the list is compiled based on searches performed by the users, and the position of books reflects the frequency searches performed after "The Godfather" was searched for. It should be noted, that this list contains three books from our experiment. The swapped position of "The Family Corleone" and "Fools Die" is due to

[54] http://www.amazon.com/
[55] http://www.google.ca/insidesearch/features/search/knowledge.html

the fact that our approach assigns high importance to an author. In the Google Knowledge Graph the importance of features does not exist. Also, the Amazon website suggests “The Sicilian” and “The Family Corleone” to buyers of “The Godfather”. This emphasizes the high similarity of these books.

Table 8.4 Results of searching for the book “The Godfather”

Our approach	“The Sicilian”, “Fools Die”, “The Family Corleone”
Google Knowledge Graph	“The Sicilian”, “The Godfather returns”, “The Family Corleone”, “The Last Don”, “Fools Die”, “The Fortunate Pilgrim”, …
Amazon	“The Sicilian”, “The Family Corleone”

Table 8.5 compares the obtained similarity values of the proposed approach in the case when similarity values are averaged and weighted. In the averaged case, properties are considered to have equal importance, thus final similarity is averaged over the similarities of all properties regardless of their dominance in defining an entity. The weighted approach is the weighted sum of the similarities Eq. (8.6). The influence of recognizing importance of properties and the application of membership functions can be easily observed.

Table 8.5 Comparison of similarity measures for averaged and weighted aggregations

Pairs of Concepts	Sim^{final} - averaged	Sim^{final} - weighted
(The Godfather, The Sicilian)	0.88	0.91
(The Godfather, Fools Die)	0.73	0.73
(The Godfather, The Family Corleone)	0.53	0.62
(The Godfather, The Master and Margarita)	0.16	0.01
(Do Androids Dream, Hyperion)	0.65	0.85
(Do Androids Dream, The Godfather)	0.12	0.25
(Do Andoirds Dream, Ubik)	0.79	0.88
(Do Andoirds Dream, The Master and Margarita)	0.19	0.26

Chapter 9

9 Linguistic-based entity matching with application in Pharmacy[56]

The web becomes an overwhelmingly huge repository of data. At the same time, users demand access to the information on the web in a more natural way. In other words, users require interaction with the web using natural linguistic terms and expect human comprehensive answers. The introduction of RDF is a promising step towards significant changes how systems can utilize the web. The very nature of RDF format that ensures high interconnectivity of pieces of data creates an opportunity to process and analyze data in a different way. In this chapter, we address the problem of processing web information using fuzzy-based technologies. In particular, we adopt a linguistic representation model to determining alternatives that match a given reference with the highest possible degree and satisfying some specific criteria. The process of comparing alternatives to the reference is feature-driven while an entity is described by its features. The proposed methodology is able to deal with features of different nature and utilize comparison mechanisms suitable for each type of features. The utilization of 2-tuple allows for comparing and aggregating linguistic-based descriptions of features, especially when the reference does not specify values of features explicitly. In experiments, we show the utilization of our approach in the domain of pharmacy. The obtained results show the advantage of using the feature-based comparison process and

[56] P. D. Hossein Zadeh, M. D. Hossein Zadeh, M. Reformat, (2015) Feature-driven Linguistic-based Entity Matching in Linked Data with Application in Pharmacy, Soft Computing Journal, Springer, pp. 1-16. (Published)

linguistic aggregation procedure over results obtained using the RDF query language SPARQL (SPARQL Protocol and RDF Query Language).

In this chapter, we utilize unique aspects of RDF and develop a methodology for matching entities that are defined via attributes expressed in different formats. The emphasis is put on entities that are described with data of both numerical and symbolic nature. The proposed approach uses the concepts of fuzziness and linguistic aggregation to combine and compare entities in order to determine their similarity and satisfaction levels in the reference to the users` requirements. In particular, we explain and present the proposed approach in the following way:

- We provide a description of a basic evaluation of similarity between RDF entities (Section 9.2). It contains the principle of the process and the references to more detailed explanations. The 2-tuple representation of linguistic terms and the aggregation process based on 2-tuples are introduced (Section 9.1).

- We propose a methodology for determining an overall similarity between RDF entities that contain different types of features (Section 9.2). It is an extension of the basic similarity evaluation. The main idea of our similarity estimation process is to determine a matching level between individual types of features, map them into a fuzzy universe, and then aggregate obtained matching levels using a linguistic process to compute overall similarity. We explain the ability to deal with distributed locations of descriptions of compared entities. We use 2-tuple representation of matching levels. The application of 2-tuple allows us to cope with variety of

different linguistic-based features of entities. Also, we utilize a linguistic aggregation mechanism representing a special case of multi-criteria decision-making processes.

- The proposed approach is able to deal with entities that are defined with symbolic, ordinal, numerical or hybrid features. Each kind of feature is compared using a mechanism that is specially designed to accommodate characteristics of the feature (Section 9.4). A set of matching mechanisms are proposed and described. An example summarizing the process is presented.

- We perform an extensive case study that verifies usefulness of the method (Section 9.5). We apply the method to a problem of finding an alternative drug based on a set of criteria provided by the user. A set of queries with increased complexity and a real-life scenario query have been created and used on a number of real-world RDF datasets containing descriptions of different aspects of drugs. We compared the results obtained with our method to the results obtained from SPARQL queries.

It should be noted that the topic of finding an alternative drug and a process of comparison of drugs' features are used across the chapter to illustrate details of the proposed methodology.

9.1 Linguistic aggregation

An important step of any multi-criteria decision-making process is aggregation of individual scores representing levels of satisfaction of each criterion. Such a process is equally important in the case of finding an entity that matches multiple requirements to the highest

degree. In this process, we deal with numeric and symbolic values representing levels of satisfaction; thus, we have adopted a 2-tuple linguistic representation model proposed in [30].

The linguistic model is based on representing linguistic information as 2-tuples. It means that satisfaction values of different criteria are expressed as pairs of: a fuzzy linguistic term and a numeric value in the range [-0.5, 0.5). The reason for adopting this approach for processing linguistically represented data is twofold. First, we deal with real-life problems where information can be better presented in an approximate and qualitative form rather than a fixed and quantitative way. Second, it reduces the information loss by means of representing the information and the results of computation, i.e., aggregation, in a continuous manner.

The application of 2-tuple fuzzy linguistic representation model implies that information is represented by 2-tuples (t, α), where *t* is a linguistic term defined in the universe of discourse U, and α is a numeric value in the interval [-0.5, 0.5). The linguistic terms $T = \{t_1, t_2, \ldots, t_n\}$ defined on U represent degrees of satisfaction of a specific criterion, e.g., *low*, *medium*, and *high*. The terms are defined such that they fully express semantics of the domain. The numeric value α represents a "deviation" from the value that is a numeric center of a linguistic term. In this chapter, we use triangular membership functions for the set of seven linguistic terms as follows, Figure 9.1:

T = { t_0: *extremelyLow*(EL), t_1: *veryLow*(VL), t_2: *low*(L), t_3: *medium*(M), t_4: *high*(H), t_5: *veryHigh*(VH), t_6: *extremelyHigh*(EH) }

The detailed description of these functions is included in Table 9.1.

Table 9.1 Linguistic terms and their membership functions

EL=(0, 0, 1)	VL=(0, 1, 2)
L=(1, 2, 3)	M=(2, 3, 4)
H=(3, 4, 5)	VH=(4, 5, 6)
EH=(5, 6, 6)	

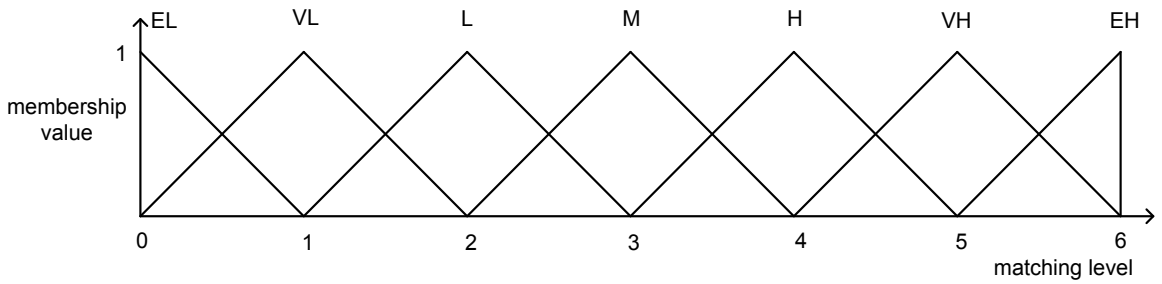


Figure 9.1 Linguistic terms t_0 to t_6 (EL - extremely low to EH – extremely high) defined in the universe of discourse *<0,6>* required for linguistic aggregation

The process of translating the result of aggregation into a 2-tuple is done in the following way: Let $\beta \in [0,6]$ represents the result of aggregation operation. The index *i* of a linguistic term, t_i, is determined as $i=round(\beta)$ and the numeric value of deviation is calculated as $\alpha = \beta - \mathrm{i}$.

For example, Figure 9.1, the linguistic term M (medium) has its numeric center equal to 3.0. In case of a value of 3.25 the "deviation" from M is 0.25. This approach allows for keeping all the original information – the translation takes place but no information is lost. The process of construing 2-tuples is presented more formally below.

So, the translation of value β into its equivalent 2-tuple is done using the following way:

$$\Delta : [0, n] \rightarrow T \times [-0.5, 0.5)$$

$$\Delta(\beta) = \begin{cases} t_i & i = \text{round}(\beta) \\ \alpha = \beta - i & \alpha \in [-0.5, 0.5) \end{cases}$$

For example, the linguistic 2-tuple of the symbolic aggregation result β = 4.2 in a linguistic term set $T = \{t_0, t_1, t_2, t_3, t_4, t_5, t_6\}$ is represented by $\Delta(\beta) = (t_4, +0.2)$.

9.2 Overview

The fundamental data format of LD – RDF – means that each piece of information is represented as a triple. This leads to a very important observation essential to the approach presented here – a single entity is a set of RDF triples with the same subject. This means that an entity is perceived as a collection of features. An example of RDF triples defining a single entity with its features is presented in Figure 9.2. The entity *Berkeley* is the subject of all triples. Each triples is a single feature describing *Berkeley*. The graphical view of such a definition of *Berkeley* resembles a star. Therefore, we use the term RDF-star to represent an entity with its features.

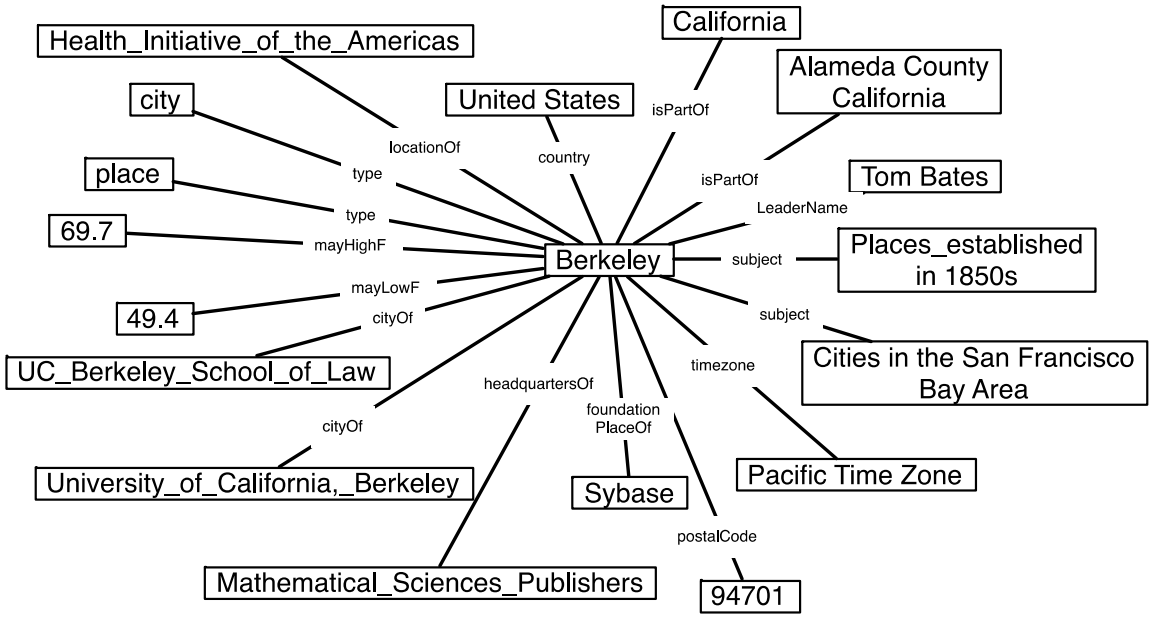


Figure 9.2 RDF triples – RDF-star – representing the entity "Berkeley"

In general, triples describing the same entity do not have to be co-located. They could be distributed among multiple RDF stores (locations). As it can be seen in Figure 9.3, an entity e_i is defined at *location A* and *location B*. Each location contains a number of features – as a feature we recognize a pair, for example <*pc*-e_r>, containing a property *pc* and another entity e_r. In other words, the entity e_i is in a relation *pc* with the entity e_r. An example could be: *Berkely* (e_i) *isPartOf (pc) California* (e_r).

The fact that we treat any entity as an RDF-star means that matching two entities can be seen as a process of comparing two RDF-stars. On one side, there is a *reference RDF*-star that represents an entity the user is interested in, and on the other side there are RDF-stars representing other entities that are available on the web. The reference RDF-star is built based

on the user's requirements regarding the entity she is looking for. Each of the requirements can create a single or multiple features of the reference RDF-star.

The process of comparison and matching is done on a feature-by-feature basis. This approach has an important advantage: it allows us not only to find exact entities (a perfect match of all features between two RDF-stars), but also not identical – similar – entities (what can be even more important in some cases).

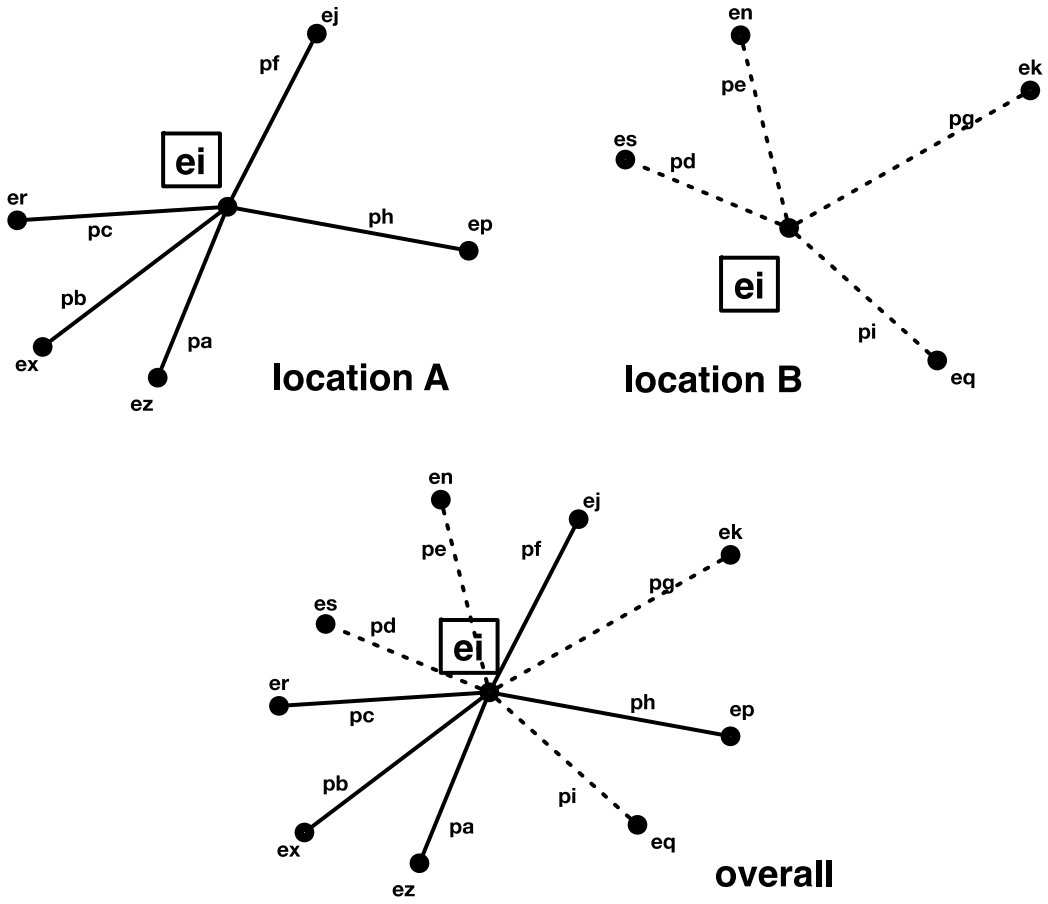


Figure 9.3 Entity e_i is defined via RDF triples stored at two different locations

In general, similarity between the reference RDF-star and any entity is obtained as an aggregation of similarities determined between equivalent features. This can be represented with:

$$\mathrm{sim}(\mathrm{e_j},\mathrm{e_{ref}}) = \mathrm{AGR}\left(\mathrm{sim}^{\mathrm{prop}}_{\forall \mathrm{p_i} \in \mathrm{P_{ref}}}(\mathrm{e_j^{pi}},\mathrm{e_{ref}^{pi}})\right) \tag{9.1}$$

where e_{ref} is a reference entity (RDF-star), e_j is an entity which is compared to e_{ref}, $sim^{prop}(.)$ is a similarity evaluation function (Section 9.3), and e_j^{pi} and e_{ref}^{pi} are features of the same type p_i. P_{ref} is a set of different features identified for comparison, while AGR can be any type of an aggregation operator. Similarity calculated in such a way is an indication of a closest possible match of the search for entity e_{ref} to any other entity.

A very important aspect of the presented approach is its high flexibility regarding different types of features the search-for-entity can have, as well as very adaptable process of comparison of these features. In reality, features of the reference RDF-star could be symbolic, discrete or numeric, and the processes of comparison of features could involve multiple steps and approaches. A comparison method would depend on types of features and mechanisms that have to be used for the features of specific types – each type of feature could invoke a different comparison process. In such a scenario, we can talk about context-based similarity. Context would mean here specific types of features, and similarity could be evaluated just taking features related to a specific context into consideration. Such a situation is presented in Figure 9.4. It shows two entities e_i and e_j that are compared based on their features <*pd-…*>, <*ph-…*>, and <*pi-…*>. Each of them is compared using a different method. Different types of features and comparison methods associated with them are described in Section 9.4.

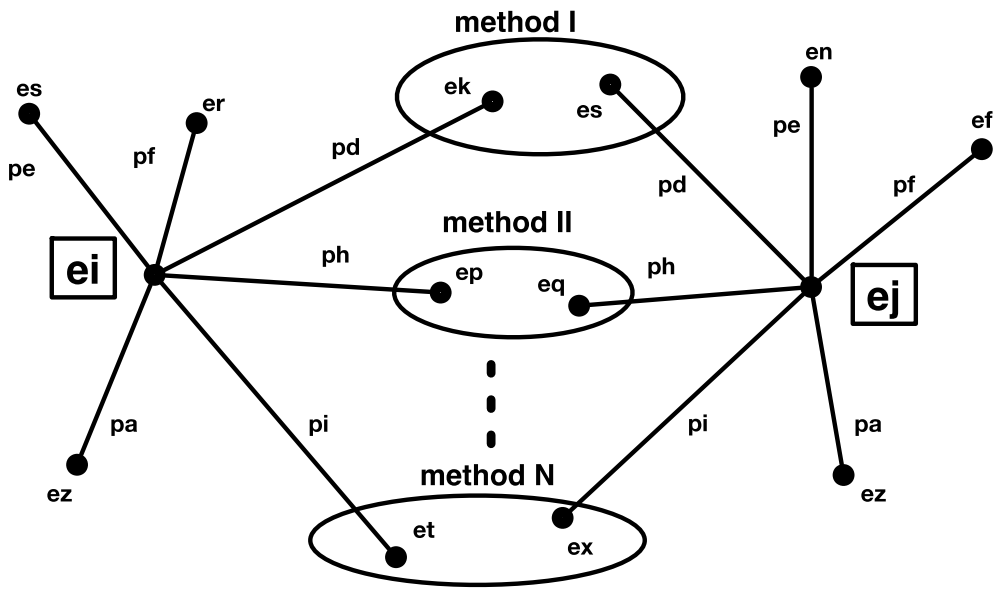


Figure 9.4 A comparison process of e_i and e_j based on three different features p_d, p_h and p_i as well as their associated comparison methods

9.3 Formulation of the matching technique

The following subsection provides a formal description of the proposed method. It fuses the RDF-based similarity evaluation with linguistic aggregation.

Let $\mathrm{sim}(\square)$ be a function calculating the similarity value in [0,1] between an entity $e_j \in \{e_1, e_2, \ldots, e_m\}$, i.e., a set of considered entities, and a reference entity $\mathrm{e}_{\mathrm{ref}}$. The function is defined as:

$$sim(e_j, e_{ref}) = \Delta\left(AGR\left(sim^{prop}_{\forall p_i \in P_{ref}}(e_j^{pi}, e_{ref}^{pi})\right)\right) \tag{9.2}$$

As it can be seen, $sim^{prop}(.)$ is a function that evaluates the context similarity between e_{j} and $\mathrm{e}_{\mathrm{ref}}$ over $p_i = \{p_1, p_2, \ldots, p_n\} \in P_{ref}$, where P_{ref} is the set of properties describing the reference entity $\mathrm{e}_{\mathrm{ref}}$, and $|P_{ref}| = n$ is a number of properties.

In this chapter, the context similarity is calculated for each property separately by considering their semantics. Applying the 2-tuple fuzzy linguistic modeling, the similarity values calculated by $\mathrm{sim}^{\mathrm{prop}}(\square)$ are in the form of a linguistic term and a numeric value: (t_i, α_i), where $t_i \in T$ and $\alpha_i \in [-0.5, +0.5)$. After obtaining such values, Eq. (9.2) has the form:

$$sim(e_j, e_{ref}) = \Delta\left(AGR\left((t_1, \alpha_1), (t_2, \alpha_2), \ldots, (t_n, \alpha_n)\right)\right) \tag{9.3}$$

Applying the inverse fuzzy linguistic representation function Δ^{-1}, the 2-tuples are transformed into their equivalent numeric values, $\beta \in [0, n] \subset \Re$.

The weighted average operator [1] is used to aggregate the constituent similarity values, β, according to their associated importance. Since each of the obtained similarities (β_i) is related to a particular property (p_i), we use different weights, w_i, considering the semantics of each property.

The aggregation operator, AGR, for similarity values is computed as:

$$AGR(\beta_1, \beta_2, \ldots, \beta_n) = \frac{\sum_{i=1}^{n} \beta_i \cdot w_i}{\sum_{i=1}^{n} w_i} \tag{9.4}$$

where the weights w_i are provided by the user, or calculated semi- or automatically, see chapter 8. The result of this aggregation process is an aggregated matching level β.

The obtained value β is represented through a linguistic 2-tuple:

$$\Delta\left(AGR\left(\beta_1, \beta_2, \ldots, \beta_n\right)\right) = \Delta\left(\beta\right) = (t_i, \alpha_i) \tag{9.5}$$

Recall, $sim(e_j, e_{ref})$ is computed for all items in the set $\{e_1, e_2, \ldots, e_m\}$ and returns linguistic 2-tuples representing the degree of similarity of e_j to the reference entity.

Finally, a list of entities representing *k* best matches to the reference entity is determined:

$$Sim\_final = \sigma^{k}_{\forall e_{j \in \{e1, e2, \ldots, em\}}}(sim(e_j, e_{ref})) \tag{9.6}$$

where σ is a function that returns a list of *k* entities e_j with the largest 2-tuple according to an ordinary lexicographic order [31] as explained below. Let (t_i, α_n) and (t_j, α_m) be two 2-tuples, then,

- if $i < j$ then (t_i, α_n) is smaller than (t_j, α_m)

- if $i = j$ then:

  (a) if $\alpha_n = \alpha_m$ then $(t_i, \alpha_n), (t_j, \alpha_m)$ represents the same information.

  (b) if $\alpha_n < \alpha_m$ then (t_i, α_n) is smaller than (t_j, α_m)

  (c) if $\alpha_n > \alpha_m$ then (t_i, α_n) is bigger than (t_j, α_m)

Once the ordered list of *k* “largest” 2-tuples is obtained, the results are presented to the user.

9.4 Context matching techniques

As stated previously, one of the most important advantages of the proposed method is the ability to use different methods of comparing features. In many situations they depend on the type/nature of features. In this section, we describe and explain a number of methods for computing context-based similarity for different features.

As mentioned previously, the proposed method is applied to the process of finding an alternative drug. Among several properties that describe a drug in LD datasets, we have selected only those that are critical factors for healthcare professionals in a process of choosing an alternative drug. These features are used to build an adequate reference RDF-star and perform a matching process. In this application, a list of selected criteria is as follows:

- pregnancy category
- side effect
- drug interaction
- route of administration

We argue that a particular similarity technique may not be suitable for all contexts, and a specific similarity method should be tailored for each context. In the next section, different similarity techniques for different types of criteria are explained.

9.4.1 Comparison of symbolic features: sequential case

The first type of comparison described in the chapter is a comparison between features that are expressed as symbols. Comparison of symbols can be seen as a string/keyword-based matching that is commonly used in most of searches performed on the web. However, the concept of semantic matching of features motivates us to provide customized comparison mechanisms. What is being proposed here is a method of comparing ordered symbols, i.e., a sequence of symbols where the position of a symbol in the sequence determines its importance and qualitative measure of its desirability. The closer a symbol is to the beginning of the sequence the more desirable it is. In order to provide a more pragmatic explanation of a suitable comparison mechanism we refer to our running example of identifying alternative drugs. Here, we will consider one of the drugs' important features: **pregnancy category.**

Every drug has a specific pregnancy category indicating the risk of fetal injury if used during pregnancy. United States Food and Drug Administration (FDA) pharmaceutical has established pregnancy classification of *{A, B, C, D, X}*, as shown in Table 9.2.

Table 9.2 FDA Pharmaceutical Pregnancy Categories

A	Adequate and well-controlled studies have failed to demonstrate a risk to the fetus in the first trimester of pregnancy (and there is no evidence of risk in later trimesters).
B	Animal reproduction studies have failed to demonstrate a risk to the fetus and there are no adequate and well-controlled studies in pregnant women.

C	Animal reproduction studies have shown an adverse effect on the fetus and there are no adequate and well-controlled studies in humans, but potential benefits may warrant use of the drug in pregnant women despite potential risks.
D	There is positive evidence of human fetal risk based on adverse reaction data from investigational or marketing experience or studies in humans, but potential benefits may warrant use of the drug in pregnant women despite potential risks.
X	Studies in animals or humans have demonstrated fetal abnormalities and/or there is positive evidence of human fetal risk based on adverse reaction data from investigational or marketing experience, and the risks involved in use of the drug in pregnant women clearly outweigh potential benefits.

To obtain a matching degree of this feature between an alternative drug and the reference drug built based on the user's requirement, we compare the RDF triple describing the pregnancy category as indicated by the reference drug to the RDF triple of the alternative drug. It is assumed that the pregnancy category is an ordered list *PregCat = {A, B, C, D, X}* with an ascending risk potential. Thus, any alternative drug with the lowest risk potential receives the highest matching level.

To calculate the matching degree of this property, we perform a direct mapping from the pregnancy categories to the fuzzy linguistic labels. Intuitively, categories A, C and X are mapped to Extremely High (EH), Medium (M), and Extremely Low (EL), respectively. According to the description in Table 9.2 categories are uniformly distributed such that category B resides in the

middle of A and C. Also, category D is in the middle of two categories of C and X. Therefore, the mapping from pregnancy categories into 2-tuples is as follows:

category A – 2-tuple (EH, 0)

category B – 2-tuple (H, 0.5)

category C – 2-tuple (M, 0)

category D – 2-tuple (VL, 0.5)

category X – 2-tuple (EL, 0)

The pregnancy categories and the associated linguistic labels are illustrated in Figure 9.5.

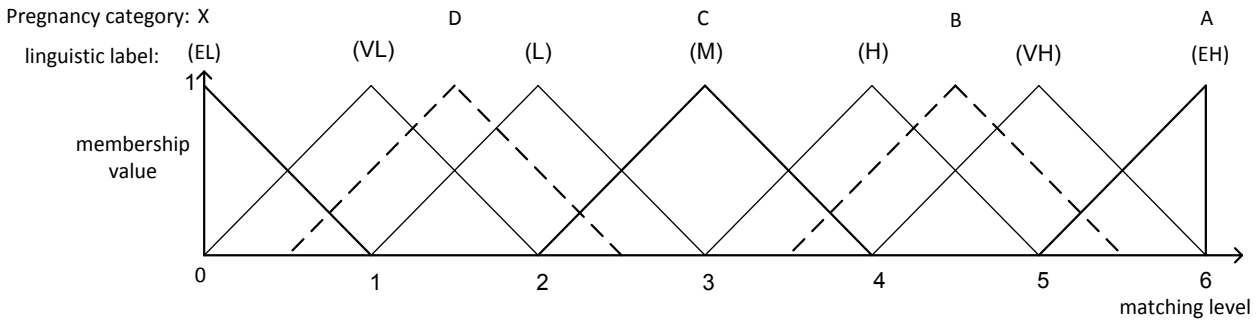


Figure 9.5 Pregnancy categories and their mappings to the linguistic labels

In this comparison process, two scenarios should be considered:

(a) user explicitly specifies a pregnancy category;

(b) user leaves the requirement regarding pregnancy unspecified, i.e, does not provide any input regarding this feature.

In scenario (a), the desired pregnancy category becomes a part of the reference drug. In order to calculate a degree of matching the following function is used:

$$sim_{preg}^{prop}(d_j, d_{ref}) = \begin{cases} SymValue(d_j^{preg}) & if\ pregnancy\ cateory\ of\ reference\ drug \\ & is \text{ less safe } than\ pregnancy\ category\ of\ d_j \\ 0 \quad otherwise \end{cases}$$

(9.7)

where d_j^{preg} is the pregnancy category of drug d_j, and *SymValue(.)* is a function that performs the mapping into the range *<0,6>*, Figure 9.5.

If the pregnancy category of the reference drug is *C* and the alternative drug's category is *D*, i.e., the reference category *C* is safer than *D* then sim_{preg}^{prop} is *0*. On the other hand, if the alternative drug's category is *B*, then *C* is less safe than *B* and sim_{preg}^{prop} is equal to *4.5* (corresponding tuple is (B, 0.0)). Such an approach allows us to differentiate between alternative drugs in case they have the pregnancy category is equal or better than the one specified by the user.

In scenario (b), the desired pregnancy category is unspecified; In this case, we have the following function:

$$sim_{preg}^{prop}(d_j, d_{ref}) = SymValue(d_j^{preg}) \quad (9.8)$$

where d_j^{preg} is the pregnancy category of drug d_j, and *SymValue(.)* is a function that performs the mapping, Figure 9.5. For example, if the pregnancy category of an alternative drug is *C* the value of sim_{preg}^{prop} is *3* (2-tuple: (C, 0.0)). Such an approach allows us to identify the best possible alternative even if the user does not provide any input regarding this particular feature.

As we can see the comparison method is fully customizable to the nature of a feature. There is not only a search for match, but also we have a chance to identify that some values are more desirable than others.

9.4.2 Comparison of symbolic features: binary case

The next category of comparison takes into consideration a number of instances of the same feature. In this case, we consider a single feature and the degree of matching depends on a number of instances of this feature. In such a situation, we cannot rely on a simple keyword matching.

The comparison process not only checks if the reference RDF-star and an alternative entity has the same feature, but it also takes into account how many of these features the alternative entity possesses. Such a situation happens when the user looks for an alternative that has a maximum possible number of features she requires.

For the case of looking for an alternative drug, a feature that can belong this category of features is **drug interaction**.

Drug interaction happens when the effect of one drug is altered by another drug. The drug interaction may lead to many harmful situations such as drug overdose, adverse side effects, serious diseases, and decrease in the effect of one or both drugs. Certain drugs can interact pharmacologically according to their mechanism of actions. Some underlying factors may increase the likelihood of drugs' side effects as well as drug interactions. They are an old age, a number of drugs taken by a patient, hepatic/renal diseases, and genetic factors.

In a real-life scenario, user identifies undesired interactions and expects to find an alternative drug that has minimum number of such undesired interactions. Computing a matching level for this type of feature can be expressed with the following formula:

$$sim_{interac}^{prop}(d_j, d_{ref}) = \prod_{i=1}^{n} interaction(d_j^{interac}, d_{ref}^{interac}) \quad (9.9)$$

where *n* is the number of undesired interactions specified in the reference drug. A function *interaction* evaluates wether an interaction does *not* exist between any two drugs. It returns 1 in case of no interactions and 0 in case of an interaction. The similarity will be the product of the obtained values. Due to a sensitive nature of this property, the computed matching level will be either one or zero. For the matching level to be one none of the undesired interactions should exist in a particular drug, otherwise the matching level is zero. The calculated matching level of 0 and 1 are mapped to the linguistic labels EL and EH (Figure 9.5), respectively.

9.4.3 Comparison of symbolic features: quantitative case

A similarity determination method that is very similar to the previous one (Section 9.4.2) is presented here. In this case, we are interested in the exact number of matching instances of the same feature. When the reference RDF-star does not specify instances of a given feature, the maximum number of instances of the feature represent the best match.

In our drug example, we identify one of the features that fits the above description: **route of administration**. The method to administer a drug to body is known as the route of administration. Some drugs should only be administered in a particular form(s). For example,

insulin cannot be given orally because when administered in this manner it is extensively metabolized in stomach before reaching blood stream, and as the consequence it would have an insufficient therapeutic effect. A variety of dosage forms exists for a single drug, since different medical conditions may demand different routes of administration. For example, for a patient with persistent nausea and vomiting it is difficult to use an oral dosage form; it may be necessary to utilize other alternate routes.

The World Health Organization (WHO) has classified route of administrations into ten categories: implant, inhalation, nasal, instillation, oral, parenteral, rectal, sublingual, transdermal, and vaginal[57]. Usually, different ways for administrating a drug are acceptable. Therefore, a matching level between an alternative drug and the reference drug can be determined based on a ratio of the number of administration routes available for an alternative drug to the number of desired routes.

$$sim_{route}(d_j, d_{ref}) = \frac{number\ of\ admin.routes\ of\ d_j\ matching\ ones\ identified\ in\ d_{ref}}{number\ of\ desired\ instances\ identified\ in\ d_{ref}} \tag{9.10}$$

If the user does not indicate a route of administration, an alternative drug with the maximum number of administration routes is selected as the best match:

$$sim_{route}(d_j, d_{ref}) = \frac{number\ of\ admin.routes\ of\ d_j}{N\ (number\ of\ possible\ admin.routes)} \tag{9.11}$$

[57] http://www.whocc.no/atc_ddd_index/

where N=10. The obtained value of similarity is in the range <0,1>. For the linguistic aggregation process it is scaled up to the range *<0,6>*, Figure 9.1.

9.4.4 Comparison of hybrid features: symbolic and numerical case

The values of this feature are a mixture of symbolic terms and numeric values. It requires a comparison procedure that can handle both numeric values in a specific range as well as multiple symbols representing different categories. The application of fuzziness allows us to map any of these values into linguistic labels defined in a suitable universe of discourse.

In our application, one of the drugs' features – **side effects** – is an example of this kind of features. Side effect is known as an unintended effect that occurs by taking a drug. Side effects have different occurring frequencies, which are classified into three categories: "post-marketing" with frequency interval of [0, 0.001], "rare/infrequent" with [0.001, 0.01], and "frequent" with [0.01, 1][58]. This attribute is taken into consideration when a second medication is required due to the patient's discomfort caused by side effects.

To determine a matching level of this property, we propose an inverse linguistic label mapping between side effect labels and the matching linguistic labels, Figure 9.6. We map low frequencies of occurring side effects into high levels of matching and vice versa. The side effect labels of "post-marketing" and "rare/infrequent" are mapped to linguistic labels of "EH" and "VH" accordingly (not shown in Figure 9.6). Due to the wide range of side effect frequency [0.01, 1] in the "frequent" category, five subsequent labels of "very low frequency", "low

[58] http://sideeffects.embl.de/

frequency", "medium frequency", "high frequency", and "very high frequency" are defined. Each of these labels is mapped to a matching linguistic label as shown in Figure 9.6.

Recall, each side effect is represented within the interval of [0, 1] that can be extracted from LD repositories. The overall side effect frequency is calculated based on the average of all side effects frequencies for a particular drug. Once the overall side effect frequency is determined, it is translated into a linguistic label defined on the matching level universe of discourse (Figure 9.6).

The original method [30] assumes that the range of β is [*0, g*] where *g* is the number of linguistic labels minus one. To ensure flexibility of the linguistic aggregation and accommodate any number of labels used to describe a required criterion, we modify the process of determining 2-tuples. This modification allows for any number of linguistic labels distributed in any way, and defined on any universe of discourse. In such a case, the deviation is determined using normalized distance between center values of membership functions associated with linguistic labels.

In the first step, we determine the interval – defined by the centers of membership functions – that contains a value of β. Let us name these centers c_L and c_H, and the linguistic labels associated with them as t_L and t_H. So, the interval $<c_L, c_H>$ contains β. Then, the value of α is calculated in the following way. We normalize the distance from the left boundary of the interval:

$$m = \frac{\beta - c_L}{c_H - c_L} \tag{9.12}$$

and the 2-tuple is:

$$(t, \alpha) = \begin{matrix} (t_L, m) & if\ m < 0.5 \\ (t_H, m-1) & if\ m \geq 0.5 \end{matrix} \tag{9.13}$$

For example, if we take the distribution of linguistic terms as in Figure 9.6. Let us determine a 2-tuples for β =0.224. The interval <c_L, c_H> that contains it is <0.16, 0.40>, and the linguistic terms are M and L equivalently. The value of m is (0.224-0.16)/(0.40-0.16) = 0.267. Because its value is lower than 0.5, the 2-tuple is (M, 0.267). Additionally, the fact that we have an inverse sequence of labels regarding matching levels the final tuple is (M, -0.267).

Next, the label is converted back into a value but this time using the linguistic label defined on the universe of discourse *<0,6>*, Figure 9.1. The value in the range <0,6> is then aggregated with values representing matching levels of other criteria. This process follows the steps defined for the linguistic aggregation, Section 9.1, and the proposed matching process, Section 9.3.

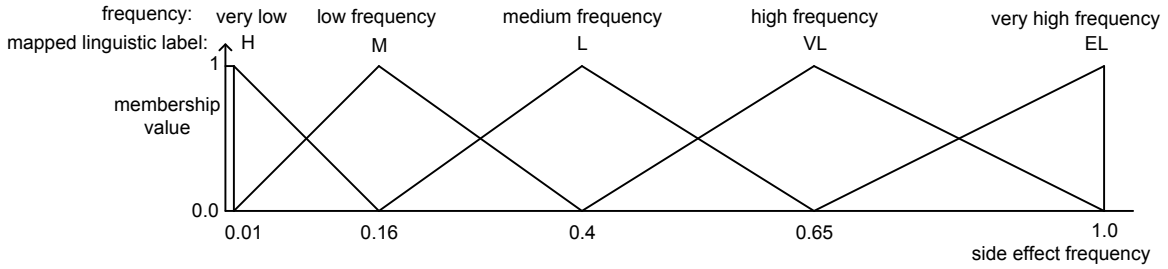


Figure 9.6 Side effect frequencies and their respective matching linguistic labels (side effect frequencies of *postmarketing* and *infrequent* are mapped into EH and VH labels, respectively)

It is worth to mention, this criterion is taken into consideration only when some side effect restrictions are indicated in the reference drug. When side effects are not specified in query, side effects of alternative drugs are evaluated for any disease interaction. More specifically, if the user is looking for a drug for alternative drugs are checked not to have minimum side effects. Thus, drugs with frequent side effects are discarded.

9.4.5 Example

Let us assume a query of the form: “Find a painkiller drug with the pregnancy category between A-C, with low side effect of vomiting, no interaction with the drug Ranitidine, and with any possible administration routes”. This query is translated into a drug reference as illustrated in Figure 9.7.

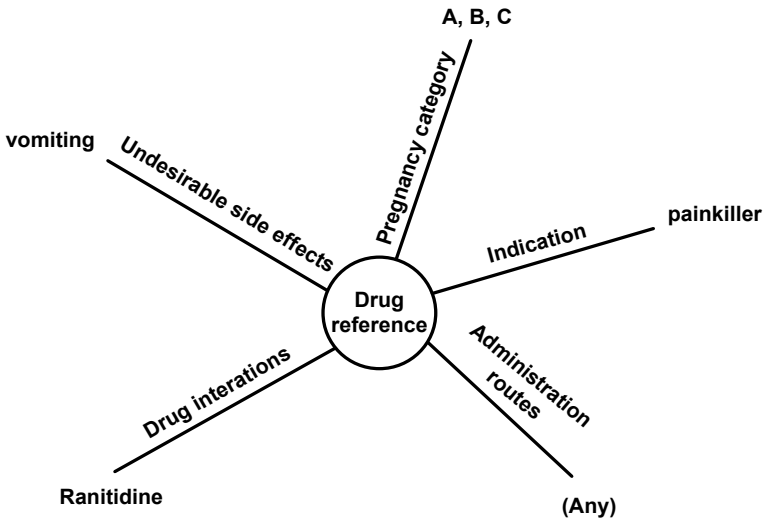


Figure 9.7 An RDF-star representing the reference drug built based on the example query

Below, we show details of the matching process, i.e., comparing the reference drug (Figure 9.7) with a set of drugs with the same indication. For the linguistic aggregation purpose, we use the following linguistic terms (Figure 9.1):

$$T = \{t_0 = EL, t_1 = VL, t_2 = L, t_3 = M, t_4 = H, t_5 = VH, t_6 = EH)$$

In the first step, a set of drugs with the same indication as the reference drug is obtained. Two such drugs are found: *Ibuprofen* and *Acetaminophen*. In the next step, features of these drugs are compared against the features of the reference drug. The information is obtained in a form of 2-tuple using a context similarity adequate for each feature. The results are shown in Table 9.3. The context similarities for different drugs' criteria are described in subsections 9.4.1 to 9.4.4. Let us take a closer look at similarity calculation of the drugs *Ibuprofen* and *Acetaminophen* to the reference drug:

- Pregnancy category: pregnancy category of drugs Ibuprofen and Acetaminophen are D and C, respectively. Based on the mapping of pregnancy categories to the matching levels (Figure 9.5), the linguistic matching of *Ibuprofen* to the reference drug is *Very Low* with deviation 0.5 and *Medium* for drug Acetaminophen with no deviation. Using Eq. (9.7), the pregnancy category matching levels are calculated as 0.0 (not meeting the minimum desired category) and 3.0 for drugs *Ibuprofen* and *Acetaminophen*, respectively.

- Side effect: side effect frequency of vomiting in *Ibuprofen* and *Acetaminophen* are 22.4% and 15% respectively. Using the mapping process described in Section 4.4 they will be mapped to matching linguistic labels of (M, -0.267) and (M, 0.067).

- Drug interaction: drugs *Ibuprofen* and *Acetaminophen* have no interaction with the drug Ranitidine; that means $sim_{interac}^{prop}$=1 for both of them. Thus, the matching linguistic label is EH according to Figure 9.1.

- Route of administration: Ibuprofen has oral, rectal, topical and parenteral routes of administration, and Acetaminophen has oral, rectal and parenteral. Based on Eq. (9.11), we calculate their matching levels 0.4 and 0.3, which are then mapped into linguistic labels of (L, 0.4) and (L, -0.2) according to Figure 9.1.

Summary of the calculations can be seen in Table 9.3.

Table 9.3 Matching levels for different criteria between drugs

Criteria/Drug	Ibuprofen	Acetaminophen
Pregnancy category	(VL, 0.5)=0	(M, 0)=3.0
Side effect	(M, -0.27)=2.73	(M, 0.07)=3.07
Interaction	(EH, 0)= 6.0	(EH, 0)= 6.0
Administration route	(L, 0.4)=2.4	(L, -0.2)=1.8
Final similarity	11.13/4=2.78 (M, -0.22)	13.87/4=3.47 (M, 0.47)

After calculating matching levels for each criterion for all the drugs, the final similarity for each drug is obtained using Eq. (9.5). For the drug *Ibuprofen* the overall score is a *Medium* with a small negative deviation while the drug *Acetaminophen* has a score of *Medium* with deviation 0.46 towards *High*. Between them, the drug *Acetaminophen* obtained the highest final score.

So, drug *Acetaminophen* is the best match to the given query. In this example, weights for all criteria are assumed to be equal.

9.5 Experimental studies

In order to demonstrate the benefits of the proposed feature-driven entity matching methodology, we show its application in the pharmacy in the process of finding an alternative drug. An alternative drug is a drug that has the same indication (target disease) as a reference drug but satisfies different criteria. This experiment also illustrates how different types of features are accommodated by adequate comparison mechanisms.

9.5.1 Explored datasets

In LD environment, there are multiple datasets related to the domain of health and medication mainly DrugBank[59], DBpedia[60], Sider[61], Diseasome[62], DailyMed[63] and LinkedCT[64]. These datasets are interconnected, i.e., they use the same terminology and contain descriptions of different aspects the same drugs. A single dataset does not provide comprehensive information about a drug, but all together the datasets constitute a thorough description of drugs. After investigating these datasets, we have selected DrugBank, DBpedia, Sider and Diseasome due to their relevance and completeness. In addition, they provide a

[59] http://wifo5-03.informatik.uni-mannheim.de/drugbank/
[60] http://dbpedia.org/About
[61] http://www4.wiwiss.fu-berlin.de/sider/
[62] http://www4.wiwiss.fu-berlin.de/diseasome/
[63] http://www4.wiwiss.fu-berlin.de/dailymed/
[64] http://linkedct.org/

heterogeneous and real large-scale data that is challenging enough to be extracted and processed. The following is a brief description of the considered datasets.

DrugBank is a repository of almost 5000 drugs translated into more than 750,000 RDF triples from the DBpedia database of almost 7000 drugs. The dataset includes drugs' chemical structure, pharmaceutical data, and drug target. The data is freely available to be accessed by web browsers and SPARQL endpoints. DBpedia contains structured data from Wikipedia while it is intensively interlinked with other data sources. It describes more than 3 million concepts and has 4800 and 2300 links to DrugBank and Diseasome datasets, respectively. LD version of Diseasome database publishes more than 4000 drug's indications along with the disease genes. DBpedia contains detailed information about all aspects of drugs such as their category, indication, chemical formula, metabolism information, etc. Sider is the LD version of the Sider database that contains information about side effects of drugs. It covers almost 1000 drugs and more than 9000 of drug-side effects pairs.

9.5.2 Queries and results

In case of drugs described in LD repositories, information related to pregnancy category of each drug is available in form of RDF. Also, drug interactions and side effects are represented in RDF, but information related to side effect frequencies and percentages cannot be found in this format. For this reason, our system extracts this information from external resources[65] [66].

[65] Lexicomp: official drug reference for the American Pharmacist Association (http://www.lexi.com/)
[66] Microdemex: http://micromedex.com/

To evaluate the effectiveness and usefulness of our method, we use a set of queries of varying complexity. The queries have been handcrafted by an expert in the clinical Pharmacy field. The set contains six different queries Query_1 to Query_6, ranging from simple to complex, and a real-life case, Table 4.3 - 4.9. Query_2 to Query_6 require combination of information from two or more datasets thus testing our system's ability to merge numbers of LD sources. To assess the ability of our method to handle linguistic terms, Query_4 to Query_6 are linguistically complex. Lastly, a real-life case is created to reflect the user's question in a real-life scenario.

First, our method translates the query into a reference drug model, and identifies properties to be matched against the reference drug. These properties are selected from a set of pre-defined properties, which are defined according to experts' opinions on the basis of common user queries. The results obtained using our approach are placed against the results obtained using the SPARQL queries. It worth noting that the aim is not to compare our method with SPARQL but to illustrate the necessity and usefulness of semantic processing of information for answering natural language based questions.

Table 9.4 Results for Query_1: Find a drug for Hypertension[67] not in pregnancy category of D and X

Our method result:	1 - Hydrochlorothiazide, Acebutolol 2 - Furosemide, Indapamide, Amlodipine, Nifedipine, Verapamil, Propranolol, Metoprolol, Bisoprolol, Clonidine

[67] High blood pressure

SPARQL result:	Furosemide, Hydrochlorothiazide, Indapamide, Amlodipine, Nifedipine, Verapamil, Acebutolol, Propranolol, Metoprolol, Monocar (Bisoprolol), Clonidine

For **Query_1: Find a drug for Hypertension not in pregnancy category of D and X**, Table 9.4, our method provides a ranked list of best-matched drugs based on the final similarity obtained by calculating a matching level using a single feature: pregnancy category. The required information related to pregnancy category is extracted from DBpedia dataset. Based on our method, two drugs Hydrochlorothiazide and Acebutolol are ranked first since their pregnancy categories are the safest among other drugs. It should be noted that drugs listed at a particular level of the ranking have equal matching degrees. For example, the final similarity values for Hydrochlorothiazide and Acebutolol are equal. As can be seen, the SPARQL query returns the drugs as long as they satisfy the pregnancy category of not being "D" and "X". For the obtained 2-tuples of similarity, see Table 9.11.

Table 9.5 Results for Query_2: Find a drug for Hypertension not in pregnancy category of D and X and no interactions with Ibuprofen

Our method result:	1- Hydrochlorothiazide 2- Indapamide, Amlodipine, Nifedipine, Verapmail, Clonidine
SPARQL result:	Hydrochlorothiazide, Indapamide, Amlodipine, Nifedipine, Verapamil, Clonidine

In order to answer **Query_2: Find a drug for Hypertension not in pregnancy category of D and X and no interactions with Ibuprofen**, Table 9.5, our method combines the information

from DBpedia and Drugbank datasets containing pregnancy categories and drug interaction information. In this query, our method retrieves a ranked list of drugs based on matching levels calculated for two properties: pregnancy category and drug interaction. When compared to Query_1, Acebutolol, Furosemide, Propranolol, Metoprolol, and Bisoprolol are omitted due to their interaction with Ibuprofen. Again, the results of our method are ranked – the drug with the highest degree of matching is ranked first, and so on. Even though all the returned drugs completely satisfy the given criteria, Hydrochlorothiazide is ranked first because it has a safer pregnancy category (B) than other drugs in the lower rank. Result from the SPARQL query returns all drugs that are equally suitable as long as they satisfy the specified criteria.

Table 9.6 Results for Query_3: Find a drug for Hypertension not in pregnancy category of D and X and no interaction with Ibuprofen and Cimetidine with oral administration route

Our method result:	1- Hydrochlorothiazide 2- Indapamide, Amlodipine, Verapamil, Clonidine
SPARQL result:	Hydrochlorothiazide, Indapamide, Amlodipine, Verapamil, Clonidine

A new criterion regarding a specific administration route as well as another undesired interaction are added in **Query_3: Find a drug for Hypertension not in pregnancy category of D and X and no interaction with Ibuprofen and Cimetidine with oral administration route**, Table 9.6. Compared to the result from Query_2, Nifedipine is discarded since it interacts with Cimetidine. Our method returns the results in which Hydrochlorothiazide is ranked first due to its higher overall matching level that is caused by the safer pregnancy category than other

drugs. So far, the items returned from our method and SPARQL are identical while they differ only in the ranking. The next set of queries is designed to show the effect of taking into account the linguistic terms in a given question.

Table 9.7 Results for Query_4: Find a drug for Hypertension not in pregnancy category of D and X and no interaction with Ibuprofen and Cimetidine and has *as many* administration routes as *possible*

Our method result:	1- Hydrochlorothiazide 2- Verapamil, Clonidine
SPARQL result:	Verapamil, Clonidine

In **Query_4: Find a drug for Hypertension not in pregnancy category of D and X and no interaction with Ibuprofen and Cimetidine and has *as many* administration routes as *possible***, Table 9.7, the result from our method returns three drugs in total divided into two ranks. Two drugs Verapamil and Clonidine are returned since they each have two routes of administration, namely (oral and intravenous) and (oral and transdermal) respectively. In contrary to the SPARQL result, Hydrochlorothiazide is also returned in our method and it is ranked first. The reason is that even though Hydrochlorothiazide has only one administration route but it has a safety pregnancy category of B, which is safer compared to C which the drugs Verapamil and Clonidine have (see Figure 9.6 in mapping of pregnancy category).

The difference in the result using our method and SPARQL is caused by the term *as many* and *as possible*. In this situation, SPARQL is inflexible and evaluates alternatives based on the number of administration routes focusing on drugs with the maximum number of routes.

Hydrochlorothiazide has a single route of administration but superior pregnancy category. This shows how considering the *overall* similarity obtained using linguistic aggregation of individual similarities can alter the answer.

Table 9.8 Results for Query_5: Find a drug for Hypertension not in pregnancy category of D and X and no interaction with Ibuprofen and Cimetidine and has a *very low* side effect of headache

Our method result:	1- Hydrochlorothiazide, Clonidine 2- Verapamil 3- Indapamide 4- Amlodipine
SPARQL result:	-- (unable to provide results)

Query_5: Find a drug for Hypertension not in pregnancy category of D and X and no interaction with Ibuprofen and Cimetidine and has a *very low* side effect of headache, Table 9.8, contains the linguistic term *very low* accompanying the side effect criterion. The reference drug has been constructed putting "headache" as an undesirable side effect (Figure 9.7). The term very low is considered due to utilization of linguistic label for evaluation of matching levels and linguistic aggregation. The results obtained with our method contain four answers. The drugs are evaluated combining the information from DBpedia, DrugBank and Sider. Side effect frequencies are mapped to linguistic labels according to Figure 9.6. Retrieved drugs are ranked with Hydrochlorothiazide and Clonidine at the very top – they have the best scores for side effects: (EH, 0.0) and (VH, -0.1) respectively, for interaction – both (EH, 0,0). For the pregnancy category Hydrochlorothiazide obtains (H, 0.5), and for the route of administration Clonidine has

the highest score of (VL, 0.2). The SPARQL query is unable to process the linguistic term and does not return any result.

Table 9.9 Results for Query_6: Find a drug for Hypertension with the safest possible pregnancy category that has no interaction with Ibuprofen and Cimetidine and infrequent gastrointestinal related side effects

Our method result:	1- Hydrochlorothiazide 2- Verapamil 3- Clonidine 4- Indapamide 5- Amlodipine
SPARQL result:	-- (unable to provide results)

Query_6: Find a drug for Hypertension with the safest possible pregnancy category that has no interaction with Ibuprofen and Cimetidine and infrequent gastrointestinal related side effects, Table 9.9, describes the question using different linguistic terms, such as safest possible, infrequent and related. An important aspect of this query is related to the process of building a proper reference drug. Specifically, scenario (b) is used for the pregnancy category criterion (Section 9.4.1), and two undesirable drug interactions are identified (Ibuprofen and Cimetidine). Situation is a bit different for the criterion, side effect. A special procedure that exploits the DBpedia structure is used here. The execution of this procedure is triggered by the term “related” and its synonyms. It uses the term gastrointestinal and extracts gastrointestinal disorders such as Nausea, Diarrhea and abdominal pain that become a part of the reference drug. These disorders are obtained from the symptoms category[68] page “Symptoms and signs:

[68] http://dbpedia.org/page/Category:Symptoms

Digestive system and abdomen"[69] using the property "dcterms: subject of". Side effect frequencies are matched against the set of linguistic labels (Figure 9.6) and the drugs with infrequent gastrointestinal side effects receive higher scores.

The results from our method are five drugs in five ranks, where Hydrochlorothiazide is found as the best match, Verapamil is the second best match and so on. Having the linguistic terms, SPARQL does not return any answer to this query.

Table 9.10 Table 10. Results for real-life scenario query: A pregnant woman with Diarrhea is diagnosed with Urinary Tract Infection, what medications are safe to recommend?

Our method result:	1- Ceftriaxone: (H, 0.2) 2- Nitrofurantoin: (M, -0.15) 3- Levofloxacin: (L, 0.2)
SPARQL result:	-- (unable to provide results)

The real-life scenario query**: A pregnant woman with Diarrhea is diagnosed with Urinary Tract Infection, what medications are safe to recommend?**, Table 9.10, illustrates a possible utilization of the proposed approach to deal with real-life issues concerning a patient. The query is translated into a reference drug, and the comparison mechanisms are invoked for each criterion (Section 9.4). The process starts with an initial query for drugs that have a disease

[69] http://dbpedia.org/page/Category:Symptoms_and_signs:_Digestive_system_and_abdomen

target of Urinary Tract Infection. The obtained list contains: Ceftriaxone, Ciprofloxacin, Levofloxacin, Nitrofurantoin, Fosfomycin and Trimethoprim/Sulfamethoxazole[70].

For the pregnancy category criterion, scenario (b) is considered, Section 9.4.1. In this case Trimethoprim/Sulfamethoxazole received the lowest score due to its pregnancy category of D. For the side effect criterion, the evaluation means avoiding any disease interaction of Diarrhea. Considering this criterion, Fosfomycin obtains the lowest score. Finally, our method recommends the following drugs: Ciprofloxacin, Nitrofurantoin and Levofloxacin with final similarity values of (H, 0.2), (M, -0.15), and (L, 0.2) respectively. As it can be seen Ciprofloxacin obtains a score a bit above High and it is quite better than the other two Nitrofurantoin and Levofloxacin.

For the above queries, quantitative analysis of the results is shown in Table 9.11. Note that the goal is not to compare our method with SPARQL but to illustrate how semantic processing of information for answering natural language based questions can benefit the user.

Table 9.11 Quantitative results

Query#	Obtained Results	
	Our method	SPARQL
Query_1	1 - Hydrochlorothiazide, Acebutolol. (H, 0.5) 2 - Furosemide, Indapamide, Amlodipine, Nifedipine, Verapamil, Propranolol, Metoprolol, Bisoprolol,	Furosemide, Hydrochlorothiazide, Indapamide, Amlodipine, Nifedipine, Verapamil, Acebutolol, Propranolol, Metoprolol, Monocar (Bisoprolol), Clonidine

[70] A combo drug that contains two components of Trimethoprim and Sulfamethoxazole.

	Clonidine. (M, 0)	
Query_2	1-Hydrochlorothiazide. (VH, 0.25) 2- Indapamide, Amlodipine, Nifedipine, Verapmail, Clonidine. (H, 0.5)	Hydrochlorothiazide, Indapamide, Amlodipine, Nifedipine, Verapamil, Clonidine
Query_3	1- Hydrochlorothiazide. (VH, 0.5) 2- Indapamide, Amlodipine, Verapamil, Clonidine. (M, 0.33)	Hydrochlorothiazide, Indapamide, Amlodipine, Verapamil, Clonidine
Query_4	1- Hydrochlorothiazide. (H, -0.3) 2- Verapamil, Clonidine. (M, 0.2)	Verapamil, Clonidine
Query_5	1- Hydrochlorothiazide, Clonidine. (M, -0.22) 2- Verapamil. (M,-0.34) 3- Indapamide. (L,0.47) 4- Amlodipine. (L,0.4)	--
Query_6	1- Hydrochlorothiazide. (M,-0.18) 2- Verapamil. (M,-0.3) 3- Clonidine. (M,-0.36) 4- Indapamide. (L,0.46) 5- Amlodipine. (L,0.44)	--
Real-life case	1- Ceftriaxone: (H, 0.2) 2- Nitrofurantoin: (M, -0.15) 3- Levofloxacin: (L, 0.2)	--

Based on the results shown in Table 9.11, our method provides a similarity measure in form of a 2-tuple with linguistic information for every given query compared to results returned by SPARQL that have no matching degrees associated. In addition, the results from our method are returned in a ranked order in contrary to results from SPARQL that has no order. It can be observed that as the amount of linguistic and semantic information in queries increases the SPARQL ability to retrieve information diminishes. As a result, SPARQL did not retrieve any result for queries 5, 6 and the real-life scenario.

9.5.3 Final arguments

Based on the results, we can imply the importance of linguistic evaluation of queries leading to the improvement of the obtained result. This shows the necessity for taking into account comparison measures aligned with human-like way of similarity assessment. Furthermore, we argue that identifying key features of an entity and evaluating each of them in their context are crucial [35].

We would like to point out that drug related LD datasets are incomplete and lack important information. In some datasets, important properties are missing such as pregnancy categories and side effects. For example, beta-blockers is another category of Hypertension drugs including Terazosin and Doxazosin that have unpublished data regarding such properties of pregnancy category and administration route. Also, links to the side effects of Doxazosin and several other drugs could not been found. This reduces the accuracy of any query answering system in LD.

Furthermore, there are issues regarding quality of medical data [49]. In our examples, it is most prominent in the case of ambiguity in pregnancy category C. In some medications, this category is safe enough to be given to a pregnant woman while in some other drugs it is not recommended at all, e.g., Levofloxacin is not recommended for a pregnant woman (see real-life case) but it has been detected as an answer by our method as this information is not documented well enough. This category is controversial and is under study to be revised in the pharmacy domain since it can be translated to different natural language terms.

Chapter 10

10 Conclusion and future work

One of the biggest issues in semantic environments is determining similarity between two concepts, which is a critical task to be used in many applications such as information extraction and retrieval, automatic annotation, web search engines, ontology matching and alignment, etc. As an example, identification of data satisfying user's request is realized by matching the query keywords to pieces of information. Such an approach and its variations are used by most of the web search engines. In this thesis, we addressed the problem of determining semantic similarity between entities in both ontology and LD environments.

Ontology defines structural organization and relations between concepts, properties and instances while it also adds semantic richness and reasoning capabilities. Similarity assessment of the entities also has a fundamental role in processing and analyzing data represented in ontology. The development of an XML-based format for representing data on the Web – Resource Description Framework (RDF) – has introduced a new way of looking at data. The data represented using this format can be seen as a network of interconnected nodes. Such representation gives the ability to look at a single piece of information and see how it interconnects with other pieces of information and what types of interconnections are used. In chapter 4, we proposed a method for determining similarity between concepts defined using ontology. In contrast to other techniques that use ontology for similarity assessment, the proposed approach focuses on the interconnections between concepts and individuals that are

concept instances. It also allows for determining similarity when specific contexts are taken into consideration. The presented approach for determining similarity between two objects has been applied in three different scenarios: concept level, individual level, and individual level within a context. In addition, the obtained degrees of similarities for concepts have been compared to a number of existing similarity estimation methods, and human judgment.

In chapter 5 of this thesis, an approach for analyzing the semantic similarity of concepts in ontology involving fuzzy sets is proposed. The proposed approach allows for determining similarity when specific contexts are taken into consideration. Fuzzy set theory is adopted for defining membership degrees of the obtained similarity values according to the level of abstraction of concepts in ontology. Realizing the effect of abstraction level of a concept in similarity measure, different fuzzy sets are defined to distinguish membership degrees associated with different abstraction levels identified by the ontology hirerarchy.

In Linked Data (LD), data is represented in a form of RDF triples. RDF triples that share the same subject are perceived as features describing an entity identified by this subject. In other words, a given entity is defined via a set of features. These features can be used to compare two entities.

In chapter 6, the proposed method constitutes an effective way of determining similarity between entities represented in RDF, the main data format of LD. Also here, the idea is based on feature-based similarity assessment model that incorporates semantics by evaluating different combination of features of entities. Elements of possibility theory and fuzzy set theory are used to better capture the semantic behind the interconnections of entities. Also,

determining similarity when specific contexts are taken into consideration is investigated. The next important research tasks include extending the proposed similarity measure to fully leverage the existing interconnections and metadata. Also, another future work is to enhance the similarity measurement by using a crawler as an information gathering support.

The work presented in chapter 7 addresses the problem of different relevance of features that should be considered when assessment between concepts within the environment of LD. Properties of an entity contribute to the overall similarity assessment based on their importance in defining a particular entity. Similarity measures related to the identified importance groups are aggregated to obtain the final similarity measure. A membership function is used to assign weights to different groups.

As the results described in the previous chapter has indicated that not all features of concepts are equally important. Therefore, we propose a novel approach to determine semantic similarity by taking into account different importance levels of properties defining concepts. In chapter 8, a novel method has been presented to identify these properties and their degree of importance using the information included in external resources. To accomplish that, the information included in Wikipedia infoboxes is utilized. Based on these importance levels, membership functions are developed. They are used to determine weights associated with levels of properties. These weights are further used to aggregate similarity measures assessed for each group of properties with the same importance. The proposed approach is deployed to evaluate similarities between several books. The RDF-based definitions of these books have been obtained from DBpedia. The results are very encouraging. They are

comparable with the lists of books identified by Google Knowledge Graph that is composed based on the users' searches, and by Amazon suggestions that are determined based on the users' purchases.

The importance of the web as a global and easily accessible data repository is unquestionable. More and more information of different types and formats becomes a part of it. At the same time, users' expectations regarding ways of utilizing the web are changing. In other words, users look for interacting with the web using human linguistic terms and expect the web would return human-like answers; requiring the information to be processed in a human-like way too. The introduction of RDF is a promising step towards significant changes aligned with users' expectations. The nature of RDF format ensures high interconnectivity of pieces of data and creates opportunities to process and analyze data in a different way.

In chapter 9 of this thesis, we propose and describe a methodology for finding alternative entities on the web once a reference entity is provided. The methodology is based on RDF data format and fuzzy-based linguistic processing of data. The process of determining the most suitable entity relies on the comparison of features. This feature-driven similarity evaluation process is highly adaptive and flexible. Different comparison mechanisms and algorithms can be applied to utilize specifics and nature of entities' features. The experimental part of chapter 9 showed that such a methodology provides more refined results that are the outcome of thorough and feature specific evaluation procedures. The information is processed in a human-like way, different types of data (numeric and symbolic) are considered, and different algorithms and techniques are used to process this data. Additionally, there is no requirement

to access all the data at one location. The developed methodology provides evidence that combining elements of fuzzy processing of data and new web technologies is a very attractive and promising way of addressing the users' needs to interact with the web.

As a future work, we suggest extension of the approach to determine similarity of complex concepts, i.e., concepts consisting of multiple basic entities; this would allow for comparison of concepts that are in a specific relationship between each other. For example entities with a relation "consist-of" could be used to define a university in a way that its faculties, departments, instructors and research groups are recognized and compared individually. Therefore, comparison of two universities would involve multiple comparisons and aggregation of the obtained individual similarity values.

Another idea is to conduct a proactive similarity assessment that would involve collecting additional information in order to resolve ambiguity or even lack of known facts. Such a process increases confidence levels in similarity, which requires a process for evaluating confidence in the assessed similarities.

Following the work in chapter 6, there is a potential for more in-depth work in the area of utilization of possibility theory, especially in aggregation of evaluations for hierarchical arrangement of concepts. In all above areas, implementation of similarity assessment algorithms in Hadoop/Spark environments for distributed execution of algorithms would be an option.

References

[1] J. Aczél, "On weighted synthesis of judgements," *Aequationes Mathematicae,* vol. 27, pp. 198-199, 1984.

[2] R. Albertoni and M. De Martino, "Semantic similarity and selection of resources published according to linked data best practice," in *On the Move to Meaningful Internet Systems: OTM 2010 Workshops,* 2010, pp. 378-383.

[3] D. Aumueller, H. Do, S. Massmann and E. Rahm, "Schema and ontology matching with COMA++," in *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data,* Baltimore, Maryland, 2005, pp. 906-908.

[4] C. Basca and A. Bernstein, "Querying a messy web of data with Avalanche," *Journal of Web Semantics,* vol. 26, pp. 1-28, 2014.

[5] Y. Bashon, D. Neagu and M. J. Ridley, "A framework for comparing heterogeneous objects: On the similarity measurements for fuzzy, numerical and categorical attributes," *Soft Computing,* vol. 17, pp. 1595-1615, 2013.

[6] T. Berners-Lee and J. Hendler, "Scientific publishing on the semantic web," *Nature,* vol. 410, pp. 1023-1024, 2001.

[7] C. Bizer, T. Heath and T. Berners-Lee, "Linked data-the story so far," *International Journal on Semantic Web and Information Systems,* vol. 4, pp. 1-22, 2009.

[8] M. Boros, W. Eckert, F. Gallwitz, G. Gorz, G. Hanrieder and H. Niemann, "Towards understanding spontaneous speech: Word accuracy vs. concept accuracy," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference On,* 1996, pp. 1009-1012 vol. 2.

[9] K. Chidananda Gowda and T. V. Ravi, "Divisive clustering of symbolic objects using the concepts of both similarity and dissimilarity," *Pattern Recognit,* vol. 28, pp. 1277-1282, 1995.

[10] O. Curé, "On the design of a self-medication web application built on linked open data," *Web Semantics: Science, Services and Agents on the World Wide Web,* vol. 24, pp. 27-32, 1, 2014.

[11] P. D Hossein Zadeh and M. Z. Reformat, "Assessment of semantic similarity of concepts defined in ontology," *Inf. Sci.,* vol. 250, pp. 21-39, 2013.

[12] P. D. Hossein Zadeh and M. Z. Reformat, "Context-aware similarity assessment within semantic space formed in linked data," *Journal of Ambient Intelligence and Humanized Computing,* pp. 1-18, 2011.

[13] H. Do and E. Rahm, "COMA: A system for flexible combination of schema matching approaches," in *Proceedings of the 28th International Conference on very Large Data Bases,* Hong Kong, China, 2002, pp. 610-621.

[14] M. Ehrig, P. Haase, N. Stojanovic and M. Hefke, "Similarity for ontologies-a comprehensive framework," in *Workshop Enterprise Modelling and Ontology: Ingredients for Interoperability, at PAKM,* 2004, .

[15] H. Eidenberger, "Evaluation and analysis of similarity measures for content-based visual information retrieval," *Multimedia Systems,* vol. 12, pp. 71-87, 2006.

[16] J. Euzenat and P. Shvaiko, *Ontology Matching.* Springer Heidelberg, 2007.

[17] P. Fankhauser, M. Kracker and E. J. Neuhold, "Semantic vs. structural resemblance of classes," *SIGMOD Rec.,* vol. 20, pp. 59-63, December, 1991.

[18] A. Farooq, M. J. Arshad and A. Shah, "A Layered approach for Similarity Measurement between Ontologies," *Journal of American Science,* vol. 6, pp. 12, 2010.

[19] G. Fenza, V. Loia and S. Senatore, "Local semantic context analysis for automatic ontology matching," in *2009 International Fuzzy Systems Association World Congress and 2009 European Society for Fuzzy Logic and Technology Conference, IFSA-EUSFLAT 2009 - Proceedings,* 2009, pp. 1315-1320.

[20] W. B. Frakes and R. Baeza-Yates, "Information Retrieval: Data Structures and Algorithms, PTR Prentice-Hall," *Inc. , Eaglewood Cliffs, New Jersey,* vol. 7632, 1992.

[21] A. Freitas, J. G. Oliveira, S. O'Riain, J. C. P. da Silva and E. Curry, "Querying linked data graphs using semantic relatedness: A vocabulary independent approach," *Data Knowl. Eng.,* vol. 88, pp. 126-141, 11, 2013.

[22] F. Giunchiglia, M. Yatskevich and P. Shvaiko, "Semantic matching: Algorithms and implementation," in 2007, pp. 1-38.

[23] P. Giuseppe, "A semantic similarity metric combining features and intrinsic information content," *Data Knowl. Eng.,* vol. 68, pp. 1289-1308, November, 2009.

[24] K. C. Gowda and E. Diday, "Symbolic clustering using a new similarity measure," *Systems, Man and Cybernetics, IEEE Transactions On,* vol. 22, pp. 368-378, 1992.

[25] W. E. Grosso, H. Eriksson, R. W. Fergerson, J. H. Gennari, S. W. Tu and M. A. Musen, "Knowledge modeling at the millennium," *Proc.KAW'99,* 1999.

[26] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowl.Acquis.,* vol. 5, pp. 199-220, June, 1993.

[27] P. A. V. Hall and G. R. Dowling, "Approximate String Matching," *ACM Comput.Surv.,* vol. 12, pp. 381-402, December, 1980.

[28] A. Harth and D. Maynard, "The Semantic Web Challenge 2012," *Web Semantics: Science, Services and Agents on the World Wide Web,* vol. 24, pp. 1-2, 1, 2014.

[29] O. Hartig, C. Bizer and J. -. Freytag, "Executing SPARQL queries over the web of linked data," in *8th International Semantic Web Conference, ISWC 2009,* 2009, pp. 293-309.

[30] F. Herrera and L. Martinez, "A 2-tuple fuzzy linguistic representation model for computing with words," *Fuzzy Systems, IEEE Transactions On,* vol. 8, pp. 746-752, 2000.

[31] F. Herrera and L. Martinez, "An approach for combining linguistic and numerical information based on the 2-tuple fuzzy linguistic representation model in decision-making," *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems,* vol. 8, pp. 539-562, 2000.

[32] G. Hirst and D. St-Onge, "Lexical chains as representations of context for the detection and correction of malapropisms," *WordNet: An Electronic Lexical Database,* vol. 13, pp. 305-332, 1998.

[33] P. D. Hossein Zadeh and M. Z. Reformat, "Fuzzy semantic similarity in linked data using the OWA operator," in *Fuzzy Information Processing Society (NAFIPS), 2012 Annual Meeting of the North American,* 2012, pp. 1-6.

[34] P. D. Hossein Zadeh and M. Z. Reformat, "Context-aware similarity assessment within semantic space formed in linked data," *Journal of Ambient Intelligence and Humanized Computing,* pp. 1-18, .

[35] P. D. Hossein Zadeh and M. Z. Reformat, "Fuzzy semantic similarity in linked data using wikipedia infobox," in *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint,* 2013, pp. 395-400.

[36] T. K. Landauer and S. T. Dumais, "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge." *Psychological Review; Psychological Review,* vol. 104, pp. 211, 1997.

[37] C. Leacock and M. Chodorow, "Combining local context with WordNet similarity for word sense identification," in *WordNet: A Lexical Reference System and its Application,* MIT Press, 1998, pp. 265-283.

[38] T. B. Lee, J. Hendler and O. Lassila, "The semantic web," *Sci. Am.,* vol. 284, pp. 34-43, 2001.

[39] G. W. Leibniz, *Philosophical Papers and Letters.* Kluwer Academic, 1975.

[40] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in 1966, pp. 707-710.

[41] Y. Li, Z. A. Bandar and D. McLean, "An approach for measuring semantic similarity between words using multiple information sources," *Knowledge and Data Engineering, IEEE Transactions On,* vol. 15, pp. 871-882, 2003.

[42] D. Lin, "An information-theoretic definition of similarity," in *Proceedings of the Fifteenth International Conference on Machine Learning,* 1998, pp. 296-304.

[43] V. Loia, S. Senatore and M. I. Sessa, "Similarity-based SLD resolution and its role for web knowledge discovery," *Fuzzy Sets Syst.,* vol. 144, pp. 151-171, 2004.

[44] A. Maedche and S. Staab, "Measuring similarity between ontologies," in *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*Anonymous Springer, 2002, pp. 251-263.

[45] L. Mazuel and N. Sabouret, "Semantic relatedness measure using object properties in an ontology," *The Semantic Web-ISWC 2008,* pp. 681-694, 2008.

[46] J. Mi, H. Chen, B. Lu, T. Yu and G. Pan, "Deriving similarity graphs from open linked data on semantic web," in *Information Reuse & Integration, 2009. IRI'09. IEEE International Conference On,* 2009, pp. 157-162.

[47] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross and K. J. Miller, "Introduction to wordnet: An on-line lexical database*," *International Journal of Lexicography,* vol. 3, pp. 235-244, 1990.

[48] J. M. Morales-Del-Castillo, E. Peis and E. Herrera-Viedma, "A filtering and recommender system for e-scholars," *International Journal of Technology Enhanced Learning,* vol. 2, pp. 227-240, 2010.

[49] J. M. Moreno, J. M. Morales del Castillo, C. Porcel and E. Herrera-Viedma, "A quality evaluation methodology for health-related websites based on a 2-tuple fuzzy linguistic approach," *Soft Computing,* vol. 14, pp. 887-897, 2010.

[50] R. M. Nosofsky, "Stimulus bias, asymmetric similarity, and classification," *Cognit. Psychol.,* vol. 23, pp. 94-140, 1991.

[51] O. Lassila and R. Swick. Resource description framework (RDF) model and syntax specification. *W3C Tech. Reports and Publications* [http://www.w3.org/TR/1999/REC-rdf-syntax-19990222]. 1999.

[52] R. Oldakowski and C. Bizer, "Semmf: A framework for calculating semantic similarity of objects represented as rdf graphs," in *Poster at the 4th International Semantic Web Conference (ISWC 2005),* 2005, .

[53] P. D. Hossein Zadeh and M. Z. Reformat, "Assimilation of information in linked data based knowledge base," in *14th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems,* 2012, .

[54] S. Patwardhan and T. Pedersen, "Using WordNet Based Context Vectors to Estimate the Semantic Relatedness of Concepts," pp. 1-8, apr, 2006.

[55] T. Pedersen, S. Patwardhan and J. Michelizzi, "WordNet:: Similarity: Measuring the relatedness of concepts," in *Demonstration Papers at HLT-NAACL 2004,* 2004, pp. 38-41.

[56] C. Pesquita, D. Faria, A. O. Falc\ao, P. Lord and F. M. Couto, "Semantic similarity in biomedical ontologies," *PLoS Computational Biology,* vol. 5, pp. e1000443, 2009.

[57] G. Pirró and J. Euzenat, "A feature and information theoretic framework for semantic similarity and relatedness," *The Semantic Web–ISWC 2010,* pp. 615-630, 2010.

[58] G. Pirró and J. Euzenat, "A feature and information theoretic framework for semantic similarity and relatedness," in *The Semantic Web–ISWC 2010*Anonymous Springer, 2010, pp. 615-630.

[59] J. R. G. Pulido, S. B. F. Flores, R. C. M. Ramirez and R. A. Diaz, "Eliciting ontology components from semantic specific-domain maps: Towards the next generation web," in *Web Congress, 2009. LE-WEB'09. Latin American,* 2009, pp. 224-229.

[60] B. Quilitz and U. Leser, "Querying distributed RDF data sources with SPARQL," in *5th European Semantic Web Conference, ESWC 2008,* 2008, pp. 524-38.

[61] R. Rada, H. Mili and E. Bicknell, "Development and application of a metricon semantic nets," *IEEE Transaction on Systems, Man and Cybernets,* vol. 19, pp. 17-30, 1989.

[62] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *IJCAI,* 1995, pp. 448-453.

[63] M. A. Rodriguez and M. J. Egenhofer, "Determining semantic similarity among entity classes from different ontologies," *Knowledge and Data Engineering, IEEE Transactions On,* vol. 15, pp. 442-456, 2003.

[64] D. Sánchez, M. Batet, A. Valls and K. Gibert, "Ontology-driven web-based semantic similarity," *J Intell Inform Syst,* vol. 35, pp. 383-413, 2010.

[65] N. Seco, T. Veale and J. Hayes, "An intrinsic information content metric for semantic similarity in WordNet," in *ECAI,* 2004, pp. 1089.

[66] N. Shadbolt, W. Hall and T. Berners-Lee, "The Semantic Web Revisited," *Intelligent Systems, IEEE,* vol. 21, pp. 96-101, 2006.

[67] T. Slimani, B. B. Yaghlane and K. Mellouli, "A new similarity measure based on edge counting," in *Proceedings of World Academy of Science, Engineering and Technology,* 2006, .

[68] B. Smith, "Ontology," *The Blackwell Guide to the Philosophy of Computing and Information,* pp. 153-166, 2003.

[69] K. Sycara, S. Widoff, M. Klusch and J. Lu, "LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace," *Auton. Agents Multi-Agent Syst.,* vol. 5, pp. 173-203, 2002.

[70] H. Tang, H. Maitre and N. Boujemaa, "Similarity measures for satellite images with heterogeneous contents," in *Urban Remote Sensing Joint Event, 2007,* 2007, pp. 1-9.

[71] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *Journal of Artificial Intelligence Research,* vol. 37, pp. 141-188, 2010.

[72] A. Tversky, "Features of similarity," *Psychol. Rev.,* vol. 84, pp. 327-352, 7, 1977.

[73] W. Wei, P. Barnaghi and A. Bargiela, "Rational Research model for ranking semantic entities," *Inf. Sci.,* 2011.

[74] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics,* Las Cruces, New Mexico, 1994, pp. 133-138.

[75] R. R. Yager, "Quantifier guided aggregation using OWA operators," *Int J Intell Syst,* vol. 11, pp. 49-73, 1996.

[76] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning-I," *Inf. Sci.,* vol. 8, pp. 199-249, 1975.

[77] L. A. Zadeh, "Similarity relations and fuzzy orderings," *Inf. Sci.,* vol. 3, pp. 177-200, April, 1971.