ONTOLOGY MATCHING BASED ON COMBINATION OF LEXICAL AND STRUCTURAL TECHNIQUES IN SEMANTIC WEB

Inaugural-Dissertation

zur

Erlangung des Doktorgrades der

Mathematisch-Naturwissenschaftlichen Fakultät

der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

Thi Thuy Anh Nguyen

aus Vietnam

Düsseldorf, August 2015

Aus dem Institut für Informatik
der Heinrich-Heine Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Referent: Prof. Dr. Stefan Conrad
Koreferent: PD Dr. Frank Gurski

Tag der mündlichen Prüfung: 29.10.2015

Life is beautiful.

(Name of the Italian film)

ACKNOWLEDGEMENTS

This thesis is created during my four years Ph.D. research at the Databases and Information Systems Institute of the Department of Computer Science at the Heinrich-Heine-University of Düsseldorf, Germany.

There are many people who have made this work possible. I would like to take this opportunity to thank all the people who helped me to complete this thesis.

First and foremost, I am truly grateful to my thesis supervisor, Prof. Dr. Stefan Conrad for his patience, guidance, critical comments throughout discussions, and great support for this work. He provides me a good chance to pursue a Ph.D. at Institute of Computer Science of Heinrich-Heine-University Düsseldorf. He has influenced my career by transforming my research skills, which are required of a good researcher. I also want to thank PD Dr. Frank Gurski, the second reviewer of my thesis, for his interest in my research and for his time and willingness to be in the committee as a second referee.

Secondly, I especially thanks to Sabine Freese for her administrative assistance and Guido Königstein for technical input in my Ph.D. work.

Thirdly, I wish to thank all my colleagues at the database group for their helps. I thank particularly my former colleague Sadet Alcic for his helps, patient listening, and the funny atmosphere he created during the time we shared the office. Furthermore, I extend my thanks to my colleagues Ludmila Himmelspach, Magdalena Rischka, Tim Schlüter and my new colleagues Robin Küppers, Matthias Liebeck, Michael Singhof, and Daniel Braun for all what they have done for me.

I would like to thank to the Ministry Of Education and Training of Vietnam (MOET) for the scholarship and the Vietnam University of Commercial, where I work, which helped and gave me a chance to study in Germany.

Also, I am grateful to all my friends in Vietnam and Germany who are not mentioned by names but have always been beside me whenever I needed their assistance and helps in work as well as my life.

Lastly, and most importantly, my love is with my great family, my parents, my husband, and my two younger sisters, who have not only mentally encouraged me but also made my day. Moreover, my parents are the source of my happiness.

Without their unconditional help, I cannot imagine how I could achieve my goal. I especially thank to my husband, Giang Ngoc Anh, for his continued support in all circumstances in my life, his endless energy, encouragements, sympathy, and love throughout my involvement with the work. He is also the place where I can anchor.

Thi Thuy Anh Nguyen
Düsseldorf, Germany
August, 2015

ABSTRACT

Title: Ontology Matching based on Combination of Lexical and Structural Techniques in Semantic Web

Nowadays, ontologies become the foundation of Semantic Web. The number of ontologies is increasing day by day. Researching on ontologies and its applications in various fields such as artificial intelligence, computational linguistics, computer science, e-commerce have been spreading and maturing. Actually, ontologies represent the characteristics of a specific domain. They include classes, properties, relationships, and instances. Since being different from background knowledge, languages used for expression, points of view of designers and entities are modeled in different ways, there is not one ontology matched perfectly to another one. This leads to heterogeneity between ontologies. In addition, the management of knowledge based on ontologies is necessary. Therefore, comparing, mapping, and integrating ontologies should be implemented in which the task of matching is to reduce ontology heterogeneity problem and identify the similarities between entities from ontologies. From the issue mentioned above, research communities have developed methods for ontology matching based on several aspects of similarity such as lexical, semantic, structural, and instances.

This thesis focuses on the task of ontology matching which has received many investigations in recent years. Although a lot of individual similarity measures are proposed, no ontology matching system uses only one technique to match. Normally, more than one similarity measure is used and the matching results are then combined to obtain the final alignment. Ontology matching systems give solutions to achieve the best possible matching results by using lexical-based, structure-based, semantic-based, and instances-based techniques together. The proposed methods used in these systems take into account different aspects of the similarity of entities in ontologies. In this work, ontology matching is based on our structural, lexical and semantic methods and use WordNet dictionary. In particular, we present an improvement of the lexical metric by applying information-theoretic and edit distance approaches, new structural and semantic measures.

The first contribution of this study is applying information-theoretic and edit distance methods to flexibly measure lexical similarity. Besides of improving the accuracy for string-based similarity degrees, this metric deals with some irrelevant situations. Our second approach is a novel structure-based similarity measure for automatic ontology matching. Being different from existing structural measures, this approach takes into account all of the ancestors of considered concepts. Another contribution of this research is a semantic similarity measure between nouns based on the structure of WordNet. This measure uses the WordNet dictionary as an external resource to take semantics of entities. Besides the positions of two entities relatively to the root in a hierarchy, this approach considers the relationships between these entities.

Our ontology matching solution is integrated by using weighted sum method to measures in which both sequential and parallel strategies are executed for computing similarity. After that, we will evaluate the quality of our system. Our approach is implemented on the benchmark dataset of the 2008 OAEI and then compared to the other systems. The experimental results show that our approach reaches good F-measure values and can compete with other automatic systems which do not use instances. The one-to-one or one-to-many alignments are generated in the final phase. The approaches presented in this thesis could also be applied in many application domains.

Keywords: Ontology matching, Structure, Lexical, Semantic.

ZUSAMMENFASSUNG

Titel: Ontologie-Matching basierend auf der Kombination von lexikalischen und strukturellen Techniken im semantischen Web

Ontologien, Systeme von Informationen mit logischen Relationen, bilden immer stärker das Fundament des semantischen Webs. Die Zahl der Ontologien nimmt tagtäglich zu. Die Forschung auf dem Gebiet der Ontologien und ihren Anwendungen in verschiedenen Feldern der Informatik wie künstliche Intelligenz, linguistische Datenverarbeitung, Computerwissenschaften und elektronischer Handel hat sich weit verbreitet. Hier repräsentieren Ontologien die Eigenschaften eines spezifischen Bereiches. Sie schließen Klassen, Eigenschaften, Beziehungen und Beispiele ein. Da sie von unterschiedlichen Hintergrundkenntnissen stammen, verschiedene Sprachen für ihre Umsetzung verwendet werden, Ansichten der Entwickler sich unterscheiden und Entitäten in verschiedenen Arten modelliert werden, passen gewöhnlich verschiedene Ontologien nicht zueinander. Grund sind Heterogenitäten zwischen den Ontologien. Weiterhin ist es notwendig, das auf der Ontologie basierende Wissen zu verwalten und weiterzuentwickeln. Deshalb müssen Ontologien verglichen, zusammengeführt, und angepasst werden. Wichtige Aufgaben sind, Ontologieheterogenitätsprobleme zu reduzieren und die Ähnlichkeiten zwischen Entitäten zu identifizieren, um die Ontologien verbinden zu können. Vom oben genannten Problem ausgehend haben Forscher Methoden für das Zusammenbringen von Ontologien entwickelt, die sich auf mehreren Aspekten der Ähnlichkeit gestützen, wie z. B. lexikalische, semantische, strukturelle und instanzielle.

Die vorliegende Arbeit konzentriert sich auf die Aufgabe des Ontologie Matchings, das in den letzten Jahren viel erforscht wurde. Obwohl viele individuelle Ähnlichkeitsmaße vorgeschlagen werden, verwendet kein System zum Verbinden von Ontologien nur eine Technik. Normalerweise wird mehr als ein Ähnlichkeitsmaß verwendet. Ergebnisse, die gut zusammenpassen, werden dann verbunden, um die Endanordnung zu erhalten. Ontologie-Matching-Systeme geben

Lösungen, die bestmöglich zusammenpassenden Ergebnisse durch das Verwenden lexikalischer, strukturbasierter, semantischer und beispielbasierter Techniken zu verbinden. Die vorgeschlagenen Methoden ziehen verschiedene Aspekte der Ähnlichkeit von Entitäten in der Ontologie in Betracht. In dieser Arbeit basiert die zusammengeführte Ontologie auf strukturellen, lexikalischen und semantischen Methoden und dem Wörterbuch von WordNet. Wir präsentieren insbesondere eine Verbesserung der lexikalischen Metrik, indem wir Informationen theoretisch anwenden und Abstandsannahmen bearbeiten, um neue strukturelle und semantische Maße zu erhalten. Der erste Beitrag dieser Studie ist informationstheoretisch und behandelt die Bestimmung von Abständen, um lexikalische Ähnlichkeit flexibel messen zu können. Neben der Verbesserung der Genauigkeit für zeichenbasierte Ähnlichkeitsgrade arbeitet diese Metrik mit einigen irrelevanten Situationen, die hier nur erwähnt werden sollen. Unser zweiter Beitrag ist ein neues strukturbasiertes Ähnlichkeitsmaß für das automatische Verbinden von Ontologien. Obwohl die Betrachtungsweise von bisherigen strukturellen Maßen verschieden ist, berücksichtigt die vorliegende Arbeit alle älteren Konzepte, die existieren. Ein weiterer Teil der vorliegenden Arbeit liegt in der Entwicklung eines semantischen Ähnlichkeitsmaßes zwischen Substantiven der Struktur von WordNet. Dieses Maß verwendet das Wörterbuch von WordNet als eine externe Quelle, um die Semantik von Entitäten zu bekommen. Neben den Positionen von zwei Entitäten relativ zur Wurzel in einer Hierarchie berücksichtigt die Arbeit die Beziehungen zwischen diesen Entitäten.

Unsere Lösung nutzt eine gewichtete Summenmethode, in der sowohl sequenzielle als auch parallele Strategien für die Berechnung der Ähnlichkeit durchgeführt werden. Danach wird die Qualität des Systems bewertet. Unsere Arbeit wurde auf einem Benchmark-Datenset des 2008-OAEI implementiert und dann im Vergleich zu den anderen Systemen durchgeführt. Die experimentellen Ergebnisse zeigen, dass unser Ansatz gute F-Measure erreicht und mit anderen automatischen Systemen vergleichbar ist, die keine Instanzen verwenden. Die one-to-one oder one-to-many Anordnungen werden in der Endphase erzeugt. Die in der Arbeit präsentierten Bertrachtungen konnten auch in viele Anwendungsgebiete übertragen werden.

Schlüsselwörter: Ontologie-Matching, Struktur, Lexikalisch, Semantisch.

DECLARATION

I, Thi Thuy Anh Nguyen, declare that this thesis titled: "Ontology Matching based on Combination of Lexical and Structural Techniques in Semantic Web" and the work presented in it are my own. I confirm that:

- This work was done mainly while in candidature for a research degree at the Databases and Information Systems Institute of the Department of Computer Science at the Heinrich-Heine-University of Düsseldorf, Germany and no portion of the work referred to in this thesis has previously been submitted for a degree or any other qualification at other university or at any other institution.
- Any material previously published or written by another person that is always clearly attributed in the text.
- I have acknowledged all main sources of help. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

_____

Date: August 15, 2015

Contents

List of Figures

List of Tables

1

INTRODUCTION

The goal of this chapter is to introduce the main topics of this thesis. It starts from the motivation and the problem of ontology matching in section 1.1. This chapter then describes objectives of the thesis in section 1.2. Furthermore, the methodology will be discussed in section 1.3. In section 1.4, a list of our contributions that has reported on chapters of this thesis is presented. Finally, an overview of the structure of the thesis given in section 1.5 will conclude this chapter.

1.1. MOTIVATION

Nowadays, the growth of the Internet and search engines is quite powerful; however seekers sometimes received knowledge which is not reasonable. Semantic Web is an extension of the current Web [11] in which computers can read, understand and search meaningful information in the way the human demanded. Therefore, to enrich the semantics of the web pages, ontologies have been developed. Ontologies become more and more popular and necessary in researching, using and maintaining. A variety of ontologies is created independently by different organizations and communities to satisfy user's requirements. Ontologies are also built in many different fields with different purposes. Moreover, there

1

exists a large number of ontologies which have the same subject and may contain overlapping information. The rapid increase in information sources and the growth in both the number and size of ontologies leads to a situation in which an increased heterogeneity occurs in the available information. Consequently, relevant pieces of information become harder to extract. To obtain the correct correspondences between entities in the ontologies being useful in the information exchange, more and more researchers pay attention to integration, comparison, matching between these ontologies [36].

As a result, a number of ontology matching systems basing on the similarity functions are proposed. However, the efficiency of these systems depends on how and what the similarity methods between entities are applied. In addition, a similarity measure is considered a good approach in case it extracts as many as possible similar entities. The individual matching techniques are based on features of entities, for example, names, individual, properties and structures to compute the similarity values. Accordingly, these techniques are useful and effective which also depend on specific ontology domains. In general, a single measure can perform well [132], however, it is not enough for determining the final alignment because the accuracy of results is not good for all kinds of domains [63]. For example, techniques based on lexical-based approach work well in ontologies in which class names having the same meaning are similar strings; however, they do not return satisfying final match results when class names use different strings for the same object having similar meanings (called synonym) or the same string for different objects (called polysemy). Therefore, to improve this situation, the matching systems should combine the results of several single similarity methods in order to achieve the final matching results instead of only one technique. Let us consider the following example.

Figure 1.1: The ontologies in the same domain

Fig. 1.1 shows a few pieces of information of the two sample ontologies about

Products. As can be seen in Fig. 1.1, in case only one string similarity measure is applied, the entity *Devices* in ontology O_1 does not match the entity *Accessories* in ontology O_2. However, if both semantic and structural measures are applied, the entities *Devices* and *Accessories* are matched together with higher similarity degree values than those obtained by each single method. This indicates clearly that the combined measure outperforms each individual technique.

In order to deal with these problems, a composition approach for ontology matching, which aggregates the proposed measures, does not only rely on strings and semantics of ontology entities but also takes their structures in the hierarchy, is discussed.

In fact, regarding various point of views and purposes of designers, ontologies are constructed differently. Moreover, since ontologies are created separately by different designers, the information included in ontologies in a same particular domain can be expressed at different levels of details and in different forms. Besides, in case the way in which the entities are defined is the same, these entities can not point to the same object. Therefore, a perfect match between two ontologies is in general not existing [101].

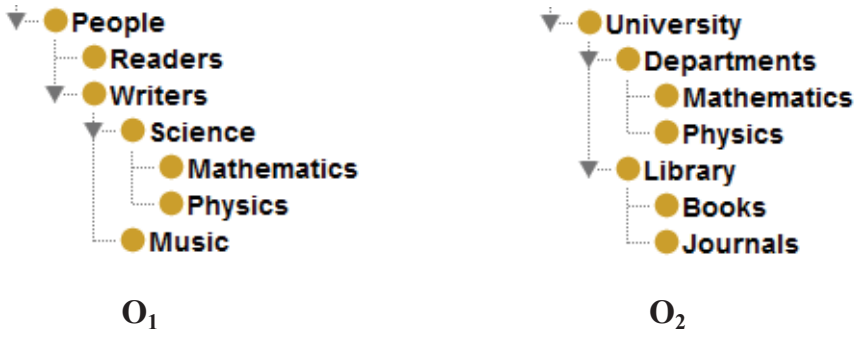

Figure 1.2: The ontologies in the different domains

Fig. 1.2 indicates that the same concept (e.g. Mathematics) in two ontologies O_1 and O_2 describes different meanings in different contexts while the similarity between two concepts *Devices* and *Accessories* in the above example (shown in Fig. 1.1) is high. As a result, the goal of ontology matching is to determine both perfectly match candidates and concepts having high confident degrees.

From the point of view described above, ontology matching is a challenging problem, however, it becomes an important issue applied in various aspects in our life. The process of the manual ontology matching is usually consumptive and expensive [81, 89]. To reduce computational costs, it is really needed to propose an automatic ontology matching solution. A number of systems applied

instance-based method, for example [26, 27, 32, 124, 144, 145]. However, using instance-based method costs a lot [81]. Therefore, in this thesis, we do not pay the attention to this approach.

1.2. OBJECTIVES

The main goal of our thesis is to contribute to the research in the field of ontology matching by combining several proposed techniques to improve the overall matching results. A structure similarity measure is introduced to match ontologies based on their structures in the hierarchy, a semantic measure based on WordNet, and a lexical measure based on the combination of information-theoretic and edit distance methods. Moreover, measures applied in this thesis can be used in various application domains. A framework is developed (called LSSOM - Lexical Structural Semantic-based Ontology Matching method), which implements the three proposed matching techniques, to align ontologies representing OWL files and return a final alignment. The performance of our proposed framework were compared and validated with other systems in terms of the match quality. The experimental results show the effectiveness of our proposed algorithm and the accuracy improved when compared to the previous ontology matching systems which do not use instances data. Finally, future directions are discussed.

In conclusion, the thesis entitled, *Ontology Matching based on Combination of Lexical and Structural Techniques in Semantic Web*, includes four key objectives. These objectives are:

- The first objective is to present the basic definitions used in this thesis, for example, ontology, ontology matching, similarity functions, and so on. Besides, the state of the art individual methods and related ontology matching systems are reviewed.
- The second objective is to investigate three similarity matching methods including lexical, semantic, and structural similarities.
- Then an approach combined similarity measures in two phases is introduced.
- Finally, the OAEI 2008 benchmark test is used for implementing, evaluating and comparing our results with the results achieved from other experiments.

1.3. METHODOLOGY

Ontology matching is often viewed as identifying similarities between the elements of two given ontologies. An ontology consists of the entities and relationships between these entities in a hierarchy. Therefore, the ontology matching problem can be considered as matching entities and relations of these input ontologies [7]. The following ideas are used for our approach. Firstly, the structural information is very important in ontologies because it contains the semantics of elements [72, 77, 90] and indicates the relationships between elements in these ontologies where these relationships are taken into account. Therefore, ontology matching based on structures in the hierarchy should be concerned. Besides, an entity can connect to many other entities to produce a complex network including parents, children, descendants, and ancestors. Consequently, our approach considers all nodes relating to the considered nodes to yield more accurately the similarities between nodes which is different from other structural systems. Secondly, the lexical techniques are usually used to discard entities in case the strings presenting these entities are almost different. These techniques are also employed in the initial phase in general. Our lexical similarity measure takes into account the common and different features between entities. Moreover, the features are chosen including the contents and positions of letters in strings. To do that, information-theoretic and edit distance methods are integrated. However, there exist entities of which the meaning is the same. On the other hand, they are depicted by the totally different strings because these entities can belong to a set of synonyms or hypernym/hyponym or holonym/meronym relations. Therefore, WordNet dictionary [75] is used in our measure to produce good results in terms of semantics which ensures that these entities are not ignored. In this semantic measure, the similarities between entities are based on synsets and relationships between these entities in the dictionary. The relationships consist of parents/children, ancestors/descendants, and direct/indirect connections. In addition, the similarity degrees depend also on the depths of the two input entities and the nearest common ancestor of these entities in the taxonomy. The proposed lexical measure is used in the initial phase of the structural process presented above. Then these individual similarities should be aggregated together. Our contribution in this thesis is an automatic composite approach, which combines three different ontology matching techniques consisting of lexical, structural and semantic-based similarities by applying weighted sum and weighted average methods. Finally, our approach is implemented on the standard benchmark data tests, the 2008 OAEI, and compared to existing matching

systems based on Precision, Recall, and F-measure values.

1.4. CONTRIBUTIONS

This thesis is generated based on results which have been presented at international conferences with reviewed proceedings and revised selected papers. The contributions of this thesis can be summarized as follows.

- The first contribution is to propose an approach for calculating the lexical similarities among the given concepts by applying the features-based and element-based measures together.
- The second contribution is to investigate a structure-based metric produced by the subsequent lexical similarity measure.
- A measure of semantic similarity of two entities in two input ontologies is proposed, which can be constructed on the features of these entities and their relevancy connections at their children.
- With an aim to address the problems of ontology matching systems, an overall framework for ontology matching is built by combining the proposed similarity measures.

The results from our publications are hereafter summarized.

1.4.1. CONFERENCE PROCEEDINGS

- Thi Thuy Anh Nguyen, Stefan Conrad: **A New Structure-based Similarity Measure for Automatic Ontology Matching**. In: The 4th International Conference on Knowledge Discovery and Information Retrieval, pages 443-449. SciTePress, 2012.

  This paper presents an approach for structure-based ontology matching, which is a sequential strategy of lexical and structural techniques. The key point to note in this approach is that all the ancestors of two concepts at different levels are considered in determining the similarity between these concepts which is different from other structural metrics. Moreover, a set of centroid concepts, which include the perfect matching concepts, was used to improve the implementation of the structure-based matching method. The experimental results of measures were compared together by using the classical measures (Precision, Recall, and F-measure).

- Thi Thuy Anh Nguyen, Stefan Conrad: **A Semantic Similarity Measure between Nouns based on the Structure of WordNet**. In: The 15th International Conference on Information Integration and Web-based Applications & Services, pages 605-609. ACM, 2013.

  One of the contributions is the proposed semantic measure. The important feature of this approach is that it uses the relationships between two considered concepts and the positions of these concepts in WordNet to calculate how similar they are. Our metric and several measures of semantic similarity were evaluated over the human judgments based on dataset of Miller-Charles by using correlation coefficients.

- Thi Thuy Anh Nguyen, Stefan Conrad: **Applying Information-Theoretic and Edit Distance Approaches to Flexibly Measure Lexical Similarity**. In: The 6th International Conference on Knowledge Discovery and Information Retrieval, pages 505-511. SciTePress, 2014.

  This paper presents another feature construction method originating from information-theoretic and edit distance. It is a fact that many existing lexical methods usually base on either ngrams or Dice's measures to produce the similarity values between strings. One of the main goals of our measure is that it takes into account common and different properties as well as the editing operations in strings motivated by Tversky and Levenshtein measures. The partial OAEI 2008 benchmark dataset and the classical measures (Precision, Recall, and F-measure) were used to compare our method with four of the common similarity metrics (Jaro-Winkler, Needleman-Wunsch, Kondrak, and Levenshtein).

- Thi Thuy Anh Nguyen, Stefan Conrad: **Ontology Matching Using Multiple Similarity Measures**. In: The 7th International Conference on Knowledge Discovery and Information Retrieval, pages 603-611. SciTePress, 2015.

  The main idea of this paper is that the lexical, structural, and semantic similarity techniques are combined to come up with an improved ontology matching system. An overall ontology matching alignment results from linear combinations by assigning different weights to the similarity components. In our approach, WordNet was employed to take semantics of entities. To evaluate the performance of matching systems, the benchmark tests of the 2008 OAEI and classical metrics for example Precision, Recall, and F-measure were employed.

1

1.4.2. BOOK CHAPTERS/JOURNAL PAPERS

- Thi Thuy Anh Nguyen, Stefan Conrad: **Combination of Lexical and Structure-based Similarity Measures to Match Ontologies Automatically**. In: Knowledge Discovery, Knowledge Engineering and Knowledge Management, volume 415 of LNCS, pages 101-112. Springer-Verlag, 2013.

  This is an extended version of the paper entitled *A New Structure-based Similarity Measure for Automatic Ontology Matching*. In this paper, the datasets was taken from five pairs of ontologies in the I^3CON 2004 to execute our system and other ones. Our obtained results were compared to the average F-Measures which are the F-Measure average values of five participants including the algorithms from Lockheed Martin ATL, INRIA, Teknowledge, AT&T, and University of Karlsruhe besides evaluating DSI and Similarity Flooding methods.

- Thi Thuy Anh Nguyen, Stefan Conrad: **An Improved String Similarity Measure based on Combining Information-Theoretic and Edit Distance Methods**. In: Knowledge Discovery, Knowledge Engineering and Knowledge Management, volume 553 of LNCS, pages 228-239. Springer-Verlag, 2015.

  This is an extended version of the paper entitled *Applying Information-Theoretic and Edit Distance Approaches to Flexibly Measure Lexical Similarity*. In this paper, a partial benchmark tests of the 2008 OAEI is used to evaluate. Moreover, a range of values of parameters α and β were recommended to yield good similarity degrees. Besides, our lexical similarity measure can be applied in different domains.

The rest of this chapter shows the outline of this thesis.

1.5. THE THESIS OUTLINE

This thesis is divided into eight chapters. A brief summary of each chapter is presented as follows. Chapter 1 introduces the motivation, objectives, methodology, and a summary of contributions of our work, as already presented. Chapter 2 reviews basic definitions and related knowledge in the ontology matching field, which are relevant to work on the following chapters discussed later on in the thesis. The most important related work that has been published on the field of ontology matching is overviewed in chapter 3. The literature review starts with three techniques including lexical, semantic, and structural similarity methods.

A set of ontology matching systems is represented in the end of the chapter 3. Chapter 4 first outlines the lexical similarity measures and the main idea of our measure in general, and then expresses a detailed description of the combination of information-theoretic and edit distance metrics. Evaluation of the experimental results is given in detail in section 4.3. Four kinds of semantic similarity measures are discussed in the section 5.1. The section 5.2 proposes an improved measure for semantics between two concepts based on positions and relationships between these related concepts in WordNet. Experiments and evaluations for this method are described in the section 5.3. An introduction of the proposed method is expressed in section 6.1. Section 6.2 provides a detail description of our structure-based approach. Section 6.3 brings an illustrative example to explain how to calculate the similarity degrees between two ontologies using our measure. This chapter also shows the experimental results and compares these results to the other ones. In section 7.1, our ontology matching composition applying lexical and structure-based techniques is reported generally. The next sections describe how the combination of the proposed measures is in order to reach a final satisfactory matching alignment, and give the performance results of our research. In the end of the chapters 4, 5, 6, and 7, discussions and the future directions of these approaches are given. The main conclusions of this thesis, scope for improvement and future trends for continuing this research are summarized in chapter 8.

2

BACKGROUND

The main purpose of the present chapter is to provide the basic background knowledge from ontologies and ontology matching in Semantic Web. For that reason, this chapter is organized as follows. Firstly, section 2.1 shows an overview of Semantic Web. Section 2.2 is started by giving definitions of ontologies and their possible application scenarios. Later on, ontology languages are discussed and a specific ontology - WordNet - is described. Furthermore, an introduction to ontology matching and application fields of the ontology matching is outlined. Ontology matching techniques are reviewed in subsection 2.3.4 which are used during the matching process. Before discussing so, the alignment for ontology matching is presented. Besides, several similarity functions are given in section 2.4. The first part of this section gives the basic notations of a similarity function. The second part contains a discussion of classification of measures concerning the problems of our approaches. After describing the benchmark tests in sections 2.5, section 2.6 presents how the classical measures can be calculated in evaluation the matching quality. Finally, this chapter closes with a short summary in section 2.7.

2.1. SEMANTIC WEB

The World Wide Web uses Uniform Resource Locators (URLs) to link web pages together. The World Wide Web contains a collection of documents and information such as images, videos, text, and other multimedia objects. On the other hand, Semantic Web is a web of linked data in which data items can connect from a source to other source by URLs. Because of the feature of linked data, information of entities can be distributed in the Web. The Semantic Web was introduced in 2001 by Tim Berners-Lee. In fact, Semantic Web is an extension of World Wide Web by adding semantic annotations to web pages in a common format which computers can read and understand. While the websites only exchange documents, the common representation formats allow Semantic Web exchange data. Hence, search engines can select the best information that users really need in a relevant time. Moreover, Semantic Web provides an easier and more intelligent way to share, combine, and reuse information.

2.2. ONTOLOGIES

Ontologies play an important role not only in Semantic Web but also in other domains such as natural language processing, biomedical informatics and so on. We present first definitions, and then languages representing the knowledge in ontologies. A special ontology, WordNet dictionary, is described at the end of this section.

2.2.1. DEFINITIONS

Ontologies in the context of the Semantic Web have been used to describe a specification of a certain domain. Ontologies represent the semantics of data, the relationships holding among objects in the real world. Thanks to them, soft agents can understand and distinguish different subjects. Because of the increase in various fields of computer and information science, numerous existing definitions for ontology are introduced depending on different contexts, modeling, and applications of ontologies. Although there is no common definition of an ontology, the definitions use the same terms. Studer et al. [125] combined the two originally ontology definitions of Gruber [45] and Borst [15] in 1998. Consequently, an ontology is considered as "a formal, explicit specification of a shared conceptualization". Some definitions relating to ontologies are mentioned below.

- An ontology consists of a collection of discrete entities (also called concepts), properties, a set of relationships between these entities in the hierarchy, and instances. Fig. 2.1 shows an illustration of the Computers ontology.

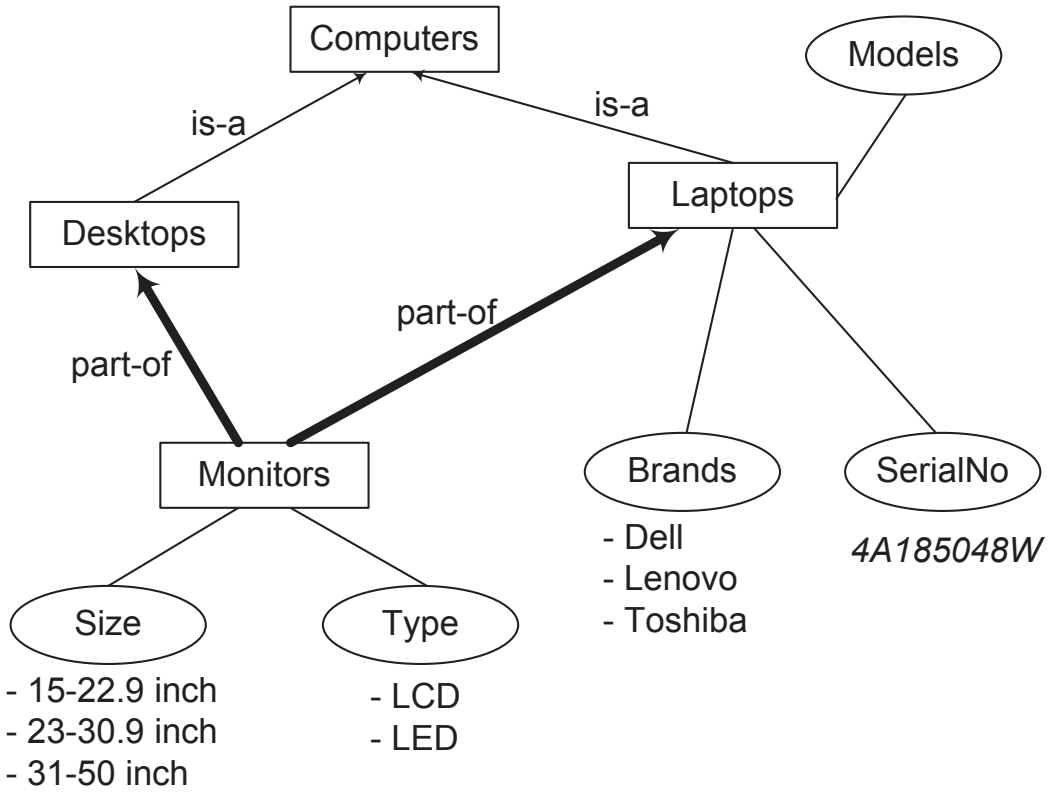

Figure 2.1: An example ontology

As can be seen, the main components of this ontology include:

- *Classes* (also called *Concepts*): a class describes a set of similar terms of the domain or task. A concept can have some subclasses and a class can have more than one superclass. For instance, *Computers* could be described as a class including subclasses such as *Desktops* and *Laptops*. A concept may have a number of properties. In Fig. 2.1, class *Laptops* has *Brands* property and class *Monitors* has *Size* and *Type* properties. Classes are depicted by rectangles.

- *Attributes* (also called *Datatype properties*): describe features of instances of the class, so a class is a set of instances with similar properties. In Fig.

2.1, each *Monitors* will have *Size* and *Type*. Attributes are displayed by ovals.

- *Relationships* are used to represent connections between elements and they are shown as arrows. These arrows point from the subclasses to their related superclasses.

- *Instances* (also called *Individuals*): instances are occurrences of a specific element and are displayed in italic strings. For example, *4A185048W* is the value of the attribute *SerialNo* and is an instance of the class *Laptops*.

Besides, ontologies usually contain added information such as datatypes and comments.

There are two kinds of ontologies: the intensional and extensional ontologies. The difference from the intensional ontologies is that the extensional ontologies consist of the instances of these ontologies [121].

One formal definition which satisfies many existing understandings of an ontology is presented as follows. An ontology is a tuple $O = (C, A, R, I)$ including the sets of ontology primitives of concepts, attributes, relations, and instances, respectively.

2.2.2. APPLICATIONS FOR ONTOLOGY

In this subsection, ontology applications are described briefly. From our point of view, ontology applications can be classified into a couple of different groups.

- Jasper and Ushold [57] grouped the ontology application domains into four categories containing neutral authoring, ontology as specification, common access to information, and ontology-based search.

- A classification of the ontology application fields comes from the work of Mizoguchi [78]. Similar to Jasper and Ushold, Mizoguchi presented scenarios for applying ontologies as specification, ontology as foundation of knowledge systematization, ontology as a common vocabulary, ontology as the help of information access, and ontology as the medium for mutual understanding.

- Staab and Studer [123] identified two main categories of the ontology application scenarios including knowledge management as well as interoperability and integration (of enterprise applications).

- Todorov [130] classified the applications into three categories: ontologies providing a common vocabulary, ontologies in support of information access, and ontologies for mutual understanding.
- Being different from the works above, Gaitanou [41] described an classification of the ontology applications consisting of six groups as Semantic Web, knowledge management, e-commerce, multimedia and graphics, grid computing, and pervasive computing environments.

2.2.3. ONTOLOGY LANGUAGES

Currently, the representation formats for ontologies that have been used on the Semantic Web consist of RDF (Resource Description Framework)/RDFS (Resource Description Framework Schema) and OWL (Ontology Web Language). These ontology languages are shown hereafter. A complete description of ontology languages can be found on the webpage of the W3 consortium [1]

- RDF/RDFS: RDF is a W3C standard recommendation for descriptions about web resources and relationships between them. The statement in RDF has three parts and is called a subject-predicate-object triple. Each of these parts is indicated by an URI. A RDF triple can be considered as a labeled graph in which the subject and object are nodes, and the predicate is a directed edge from the subject to the object. RDFS is extended from RDF with schema vocabulary such as class, property, subClassOf, subPropertyOf, range, domain. RDFS can create new properties and classes, which is different from RDF. Both RDF and RDFS are used to express and exchange metadata between applications so they produce machine-readable data. Moreover, RDF/RDFS contain semantic constraints on data.
- OWL: OWL is an extension of RDFS. It allows to add more vocabulary for defining concepts, indicating properties or relationships of concepts, the cardinality constraints, and characteristics of properties. There are three language of OWL known as OWL-Lite, OWL-DL, and OWL Full.
  - OWL Lite defines some simple restrictions for easy execution.
  - OWL DL uses Description Logic and defines some restrictions. Besides, OWL DL and OWL Lite separate classes, properties, instances, and data values.

[1] http://www.w3.org/TR/owl-ref/

– OWL Full does not define any restrictions, which is different from OWL Lite and OWL DL.

2.2.4. WORDNET

The WordNet project [75] was begun in the mid-1980s by George A. Miller. The WordNet is a freely available large lexical database of the English language containing about 155000 words. Information in WordNet is organized according to word meanings. Comparing to other traditional dictionaries, nouns, verbs, adjectives, and adverbs in WordNet are classified into synsets in which each synset contains a set of synonyms indicating a discrete concept [75]. Each sense of a word is in a synset so each word may be in various synsets. The WordNet contains over 117000 synsets. These synsets are linked together by a number of semantic relationships. As a result, WordNet is organized into hierarchies based on 25 primitive categories for nouns. These basic categories are connected to a root node. The different relationships between nouns in the WordNet consists of synonym, antonym, hypernym, hyponym, holonym, and meronym. These relationships are defined as:

- Synonym: as aforementioned, the synonyms are words that are similar or have a related meaning to another word. These words belong to a synset in WordNet. For example, *Computer* and *Computing_device* are synonyms.
- Antonym: the antonym indicates a word that means the opposite of another word. For example, *Strength* and *Weakness* are antonyms. Note that not all nouns have antonyms.
- Hypernym (as called "is-a" relation): e_1 is a hypernym of e_2 if e_2 is a kind of e_1. For example, *Laptop* is a kind of *Computer* so *Computer* is a hypernym of *Laptop*.
- Hyponym (as called "subsumes" relation): e_2 is a hyponym of e_1 if e_2 is a kind of e_1. For example, *Laptop* is a hyponym of *Computer* because *Laptop* is a kind of *Computer*.
- Holonym (as called "has-a" relation): e_1 is a holonym of e_2 if e_2 is a part of e_1. For example, *Monitors* is a part of *Desktops* so *Desktops* is a holonym of *Monitors*.
- Meronym (as called "part-of" relation): e_2 is a meronym of e_1 if e_2 is a part

of e_1. For example, *Monitors* is a meronym of *Desktops* because *Monitors* is a part of *Desktops*.

2.3. ONTOLOGY MATCHING

As already mentioned above, ontologies are applied in various domains such as natural language processing, information retrieval, e-commerce, social networks and so on. Ontologies can also be used to manage large databases. Because points of view of designers are different, ontologies representing the same pieces of knowledge are not the same. This leads to the existence of many different ontologies, which describe similar or overlapping knowledge (called heterogeneity). There are several classifications of heterogeneity [35, 64, 102, 134]. Euzenat et al. [37] divided heterogeneities into four groups as follows.

- Syntactical heterogeneity: ontologies are represented by different languages or knowledge formalisms.
- Terminological heterogeneity: entities in ontologies use different names to describe the same objects.
- Conceptual heterogeneity: the differences between modeled entities of the same domain in terms of coverage, granularity or scope.
- Semiotic heterogeneity refers to how entities are interpreted by people in a given context.

The objective of matching is to reduce heterogeneities among ontologies.

2.3.1. DEFINITIONS

Ontology matching is a process taking two given ontologies to return matching pairs (also called correspondences) between entities of these two ontologies [37]. These ontologies are considered as the source ontology (as also called referenced ontology) and the target ontology. Additionally, these ontologies can be represented in many different formats such as schemas and graphs.

A match is described as a tuple (E_1, R, E_2) where E_1, E_2 are sets of entities in the source and target ontologies, respectively, and R is an expression showing the relationships between E_1 and E_2. In the current algorithms, relationships R can be generated as *equivalence* ($=$), *more general* ($\sqsupseteq$), *less general* ($\sqsubseteq$), *disjointness* ($\perp$), and *idk* ($Idonotknow$).

2.3.2. APPLICATIONS FOR ONTOLOGY MATCHING

According to Euzenat [37], basing on the point of view of technology, ontology matching is applied in the following different scenarios.

- Ontology engineering refers to the designing, implementing and maintaining ontology-based applications of users. Instead of building the desired ontology from available information, an approach should be done to reuse or combine suitable ontology sources. To do that task, ontology matching algorithms need to identify the relevant distributed ontologies, similarities of entities in these ontologies, and differences of entities modified from multiple versions of an ontology [51, 62, 88, 92, 93, 109].
- Information integration: in case heterogeneous ontologies are used to implement the common tasks, one of the most important tasks is to find the corresponding matches among the entities in these different ontologies and then integrated together [20, 28, 43, 117, 135].
- Linked data: on the web of data, it is necessary to link the related data sets from different sources. However, the data sets are expressed by various types, for example, heterogenous schemas or ontologies. Therefore, finding correspondences between these data is needed for discovering, sharing, and connecting [55, 94, 95].
- Peer-to-peer information sharing: to exchange and share information between different peers, for example, file sharing systems, entities in these parties have to be matched to identify correspondences between them [12, 122].
- Web service composition: the main task of the combination of several services is to obtain a specific goal. In fact, web services are represented by different languages. Therefore, matching the entities in service descriptions is needed. [42, 108].
- Autonomous communication systems: these systems interact with each other by exchanging messages. Therefore, the content of messages is translated and then matched to help in the understanding of these models together [9, 140].
- Navigation and query answering on the web: thanks to the matching process, search engines transfer each user's query into the concepts of the relevant available ontologies and then return the reasonable results.

A number of studies related to solutions and techniques have been proposed to measure similarities between two entities, for example, lexical similarity, semantic similarity, and instances similarity. Ontology matching techniques, which are directly relevant to our approach, are discussed separately in subsection 2.3.4. Ontology matching stands for searching and identifying semantic correspondences among the entities of two given ontologies and the relations that hold between them and decreases heterogeneity between two ontologies. The process of ontology matching produces the ontology alignment, which will be discussed below.

2.3.3. ONTOLOGY ALIGNMENT

The similarity measures take two ontologies as inputs and then return a set of matches. This set of matches is called the alignment. An ontology alignment, denoted as A, includes a set of the concepts of a source ontology O_1 connect to a set of the concepts of a target ontology O_2. To produce the alignment, the ontology matching process is illustrated as follows [119]:

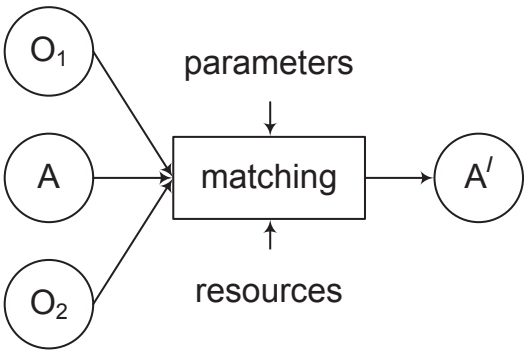

Figure 2.2: The general ontology matching process [119]

In Fig. 2.2, O_1 and O_2 are the input ontologies. To achieve the alignment A' of these ontologies, the matching algorithms can deal with the parameters including the input alignment A, the matching parameters such as weights and thresholds, and external resources such as WordNet, Euronet. The weights and thresholds can be chosen automatically or manually. An alignment brings similarity values between entities such that these similarity degrees can be normalized to obtain values in the [0,1] range. Note that one or more entities from the first ontology can match to one or more entities of the second one. Therefore, three possible results including one-to-one, one-to-many, and many-to-many alignments are considered as the *cardinality* in ontology matching.

2.3.4. ONTOLOGY MATCHING TECHNIQUES

This section reviews current ontology matching techniques. Because of the different types of heterogeneity of ontologies, there are some classifications of ontology matching techniques discussed in [23, 25, 30]. The two classifications of ontology matching techniques related to our research, which are provided by Rahm and Bernstein [104] and Euzenat and Shvaiko [37], are laid out. The latter is based on the former one.

Fig. 2.3 shows a classification of schema matching approaches proposed by Rahm and Bernstein [104]. As can be seen, matching systems are classified into two main groups: individual and combining matchers in which composite matchers combined of several individual matching techniques. For individual matcher, systems are considered either schema-based or instance/contents-based. Schema-based systems take into account the structure of the ontologies while instance/contents-based systems use the information of instances or the content of these ontologies. Moreover, schema-based approaches can apply both element-level and structure-level techniques in contrast to instance/contents-based approaches which only use element-level technique. Finally, the constraint based approaches use relationships between entities, for example, type similarity, graph matching and value pattern, so they belong to element-level and structure-level techniques. Linguistic approaches base on textual descriptions of entities, for instance, names and comments, and only belong to element-level techniques.

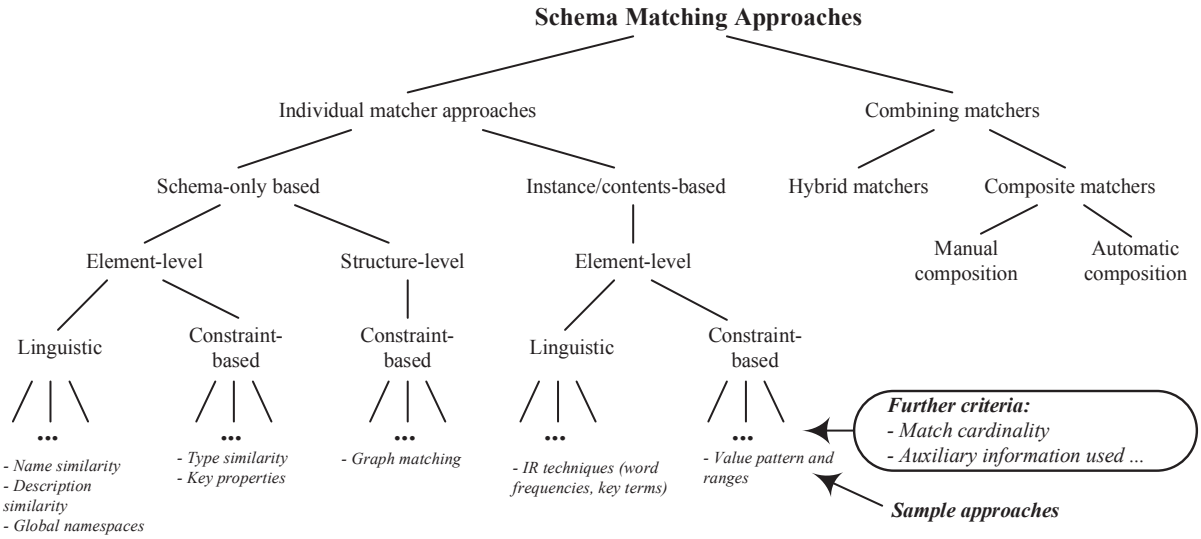

Figure 2.3: Schema matching approaches [104]

Regarding to Rahm and Bernstein's classification, Euzenat and Shvaiko categorized ontology matching approaches in more details. As can be seen in Fig. 2.4, granularity/input interpretation classification includes of element-level and structure-level versus origin/kind of input classification consists of content-based (as called internal matching) and context-based (as called external matching). Element-based, structure-based, and context-based techniques are divided into two distinct groups: semantic and syntactic. For the internal matching, it is a composite of four classes of semantic, extensional, structural, and terminological techniques. Moreover, both two classifications granularity/input interpretation and origin/kind of input contain shared classes of matching techniques which belong to the layer of concrete techniques.

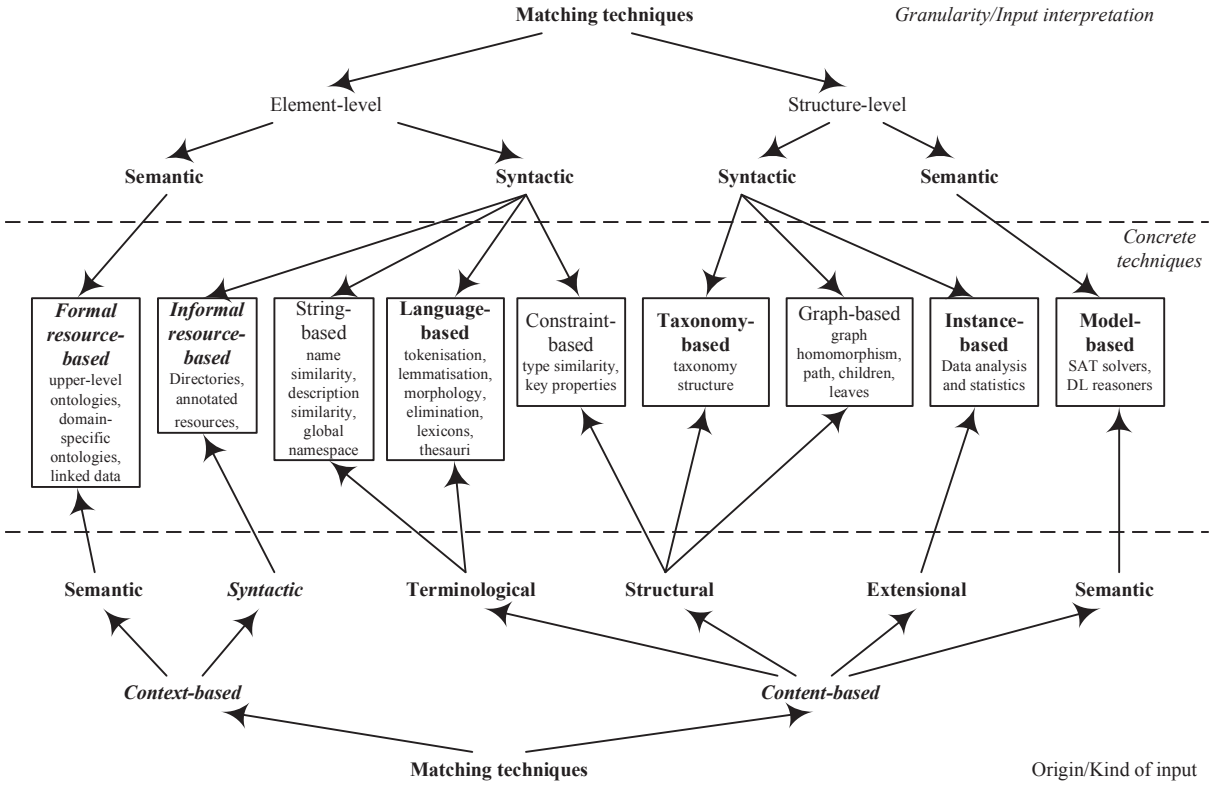

Figure 2.4: Ontology matching techniques [37]

The main characteristics of each of these techniques connected to our subject are outlined below.

- Element-level techniques do not take into account the relationships between entities and instances in ontologies. These techniques are classified into five measures: lexical-based, language-based, constraint-based,

informal resource-based, and formal resource-based techniques. Lexical-based, language-based, and constraint-based techniques are helpful in case the same concepts are expressed by highly similar strings and attributes. Otherwise, external resources should be use to improve matching results. These techniques are discussed as follows.

  - Lexical-based (also called string-based) techniques are usually applied for comparing names, labels, and comments of ontology entities in order to find the similarities between them. These objects can be considered as chains of letters. Prefix, suffix, edit distances, and n-gram similarity are lexical-based methods used in many matching systems. In these techniques, the more similar the strings are, the more likely they are to denote the same concept.
  - Language-based techniques determine the similarities of concepts based on natural language processing techniques. They consider names, labels, and comments of ontology entities as words or phrases in the natural language.
  - Constraint-based techniques consider internal constraints of entities. They calculate the similarity between entities based on data types, properties, cardinality, ranges, and domains of these entities.
  - Informal resource-based techniques use informal resources such as pictures to determine the equivalent of entities in ontologies.
  - Formal resource-based techniques depend on external resources, for example WordNet dictionary and other ontologies.

- Structural-level techniques: in contrast to element-level techniques, structural-level techniques reflect on relationships between entities in the hierarchical structure. The main well-known structure-level techniques consist of graph-based, taxonomy-based, instance-based, and model-based techniques.

  - Graph-based techniques consider the given ontologies as labeled graphs where vertices indicate classes and edges correspond to the relationships between the pairs of vertices.
  - Taxonomy-based techniques: a taxonomy consists of a set of entities organized into a hierarchical structure which is a special case of graph-based techniques. However, these methods focus only on the “is-a” relationship.

– Instance-based techniques: instances of entities are used to determine the similarities between these entities.

– Model-based techniques decide if entities base on similar semantic interpretation of these entities.

A ontology matching system can be a single-technique or multi-technique system. A single match measure, such as language-based method, can not provide a good solution for a whole matching task, so a number of the single measure should be integrated to improve results [37, 91]. Therefore, some matching techniques mentioned above are combined by using different strategies in a system instead of an individual similarity technique to increase the matching quality. A sigmoid function usually is employed to combine these strategies [69] in which the weights are automatically or manually determined. The matching strategies consist of sequential and parallel compositions. In the sequential composition, a matcher uses the alignment created in the matching process before producing the final alignment (Fig. 2.5).

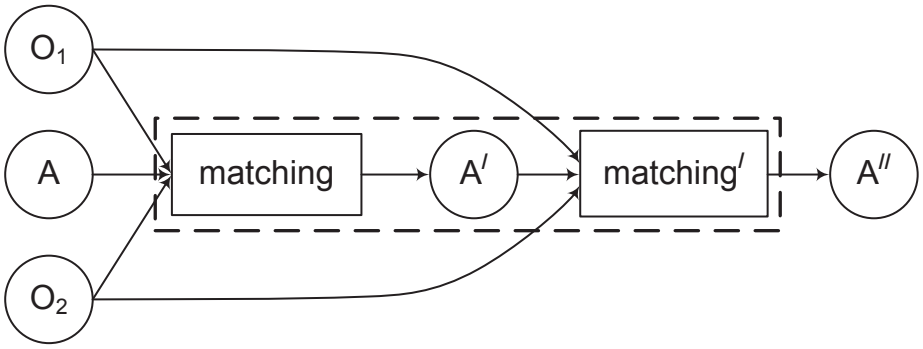

Figure 2.5: Sequential composition strategy [37]

The parallel composition is a strategy to allow two or more matchers to execute independently and then aggregate their results (Fig. 2.6)

In addition, the matching process can use the auxiliary information such as dictionaries, thesauri, and input alignments to obtain semantics of concepts, synonyms, relations and so on, which improve the final alignments. For a more detailed discussion of the ontology matching techniques, see [37].

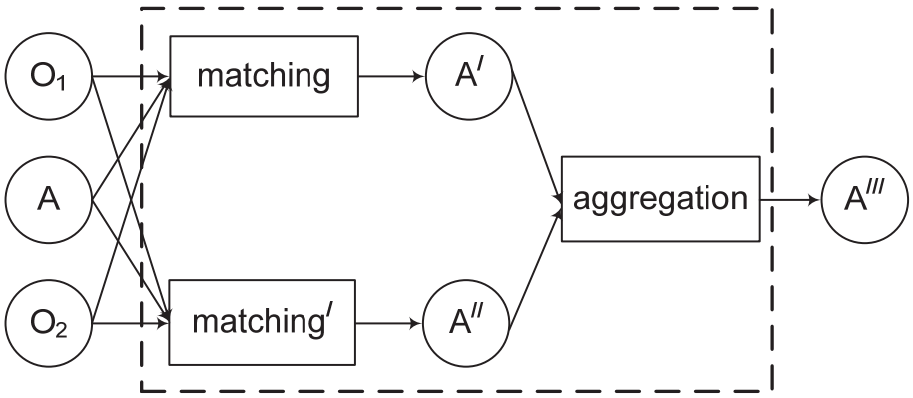

Figure 2.6: Parallel composition strategy [37]

2.4. SIMILARITY FUNCTIONS

Ontology matching can be considered as identifying similarities among the entities of two given ontologies, by applying similarity functions [47]. In the beginning of the section, definitions of a distance function as well as similarity, dissimilarity, and relatedness measures are introduced, which will be needed in the thesis. After that, an overview of these measures for classification will be given.

2.4.1. DEFINITIONS

A distance function is a function determining a distance between two entities of a set. The distance function assigns a given entity pair to a real number. Let E be a set of entities and $\mathbb{R}$ be a set of real numbers. For all $e_1, e_2, e_3 \in E$, a distance function on the set E is a function $f : E \times E \rightarrow \mathbb{R}$ satisfying the four following properties [37]:

1. $f(e_1, e_2) \geq 0$ (positiveness)
2. $f(e_1, e_2) = 0$ if and only if $e_1 = e_2$ (definiteness)
3. $f(e_1, e_2) = f(e_2, e_1)$ (symmetry)
4. $f(e_1, e_3) \leq f(e_1, e_2) + f(e_2, e_3)$ (triangle inequality)

- Dissimilarity: a dissimilarity function satisfies three following properties [37]:
  - $f(e_1, e_2) \geq 0$ (positiveness)

  - $f(e_1, e_1) = 0$ (minimality)
  - $f(e_1, e_2) = f(e_2, e_1)$ (symmetry)

  Dissimilarity is related to the distance between two entities which is the inverse of the similarity.

- Similarity Measures: the following three properties hold for a similarity measure [37]:

  - $f(e_1, e_2) \geq 0$ (positiveness)
  - $f(e_1, e_1) \geq f(e_2, e_3)$ (maximality)
  - $f(e_1, e_2) = f(e_2, e_1)$ (symmetry)

  Note that a smaller dissimilarity value indicates a greater similarity between each of pair of entities in the set E. To determine the semantic similarities between entities, similarity measures uses synonyms and the "is-a" (hyponym/hypernym) relationship and are considered as a specific case of relatedness measures.

- Relatedness Measures: relatedness measures consider relations between entities in the hierarchy. Besides the hyponym/hypernym relations, these measures also use different types of relationships, such as meronymy and antonymy [16].

In the following subsection, a classification of measures will be explained.

2.4.2. CLASSIFICATION OF MEASURES

In the literature reviews, measures are usually normalized to return the similarity values in the range of [0,1] for a pair of entities of two input ontologies. These metrics can be divided into three distinct groups: string-based, language-based, and structural techniques. These techniques are briefly presented in hereafter.

- String-based measures: string-based measures consider a string representing the name of an entity as a sequence of characters. These measures focus on how sequences are similar to another one to find out the similarities between entities.

- Language-based measures: according to a linguistic point of view, the structures of strings can be totally different while they have the similar meaning,

for example, "magician" and "wizard". Therefore, to calculate the similarity between two entities based on linguistic, a resource is used for this purpose. The most popular lexicon in English is WordNet dictionary [39, 75].

- Structure-based measures: string-based and language-based measures can yield inaccurate results because these measures concentrate on the names of entities while the relationships between entities in ontologies are ignored [81]. Consider the following illustrate example.

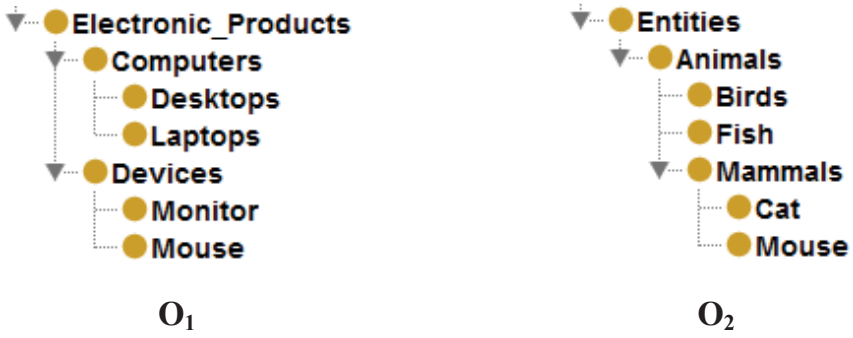

Figure 2.7: Another example ontologies

In case string-based and language-based measures are applied, the entities "Mouse" of two ontologies (see Fig. 2.7) are totally similar. However, these entities are dissimilarity because they belong to two given ontologies in different domains. Therefore, the structural method is implemented to determine the similarity degrees of entities.

In the next section, the well-known benchmark tests are overviewed.

2.5. BENCHMARK TESTS

Several ontology matching systems using different techniques are developed for a lot of purposes. Besides matching functions, benchmark tests are used as inputs for comparing and evaluating the matching quality of these systems. There are many well-known benchmark data sets to test matching ontology systems as well as the pairs of terms which were scored by experts for similarity measures [40, 76, 111, 146]. Three benchmarks applied in the scope of this thesis are presented in the following subsections.

2.5.1. R&G AND M&C

For two R&G and M&C benchmark tests, the similarity measurements of word pairs with human-assigned similarity scores are on a scale from 0 (no similarity) to 4 (totally similar) according to the similarity of meaning. The larger relatedness between the pair of terms is, the higher the score is.

The R&G dataset including a collection of 65 pairs of English nouns was carried out by Rubenstein and Goodenough in 1965 [111]. This dataset was evaluated by 51 judges (all native English speakers) to determine the relatedness values of all pairs of nouns.

In 1991, Miller and Charles (M&C) [76] chose a subset of 30 pairs in the above dataset to experiment again. In 30 pairs of Miller and Charles dataset, there are 10 pairs at high level (the values between 3 and 4), 10 pairs at intermediate level (the values between 1 and 3) and 10 pairs at low level (the values between 0 and 1). This dataset contains judgements from 38 human subjects (all native English speakers). Obtained results by using this dataset are only concentrated meaning of terms while semantic relationships are ignored.

The correlation between experimental results by using M&C and R&G datasets is high [143]. It means these human judgements can be considered as stable and good datasets for evaluation and comparison between methods [100].

2.5.2. I^3CON 2004

I^3CON (The Information Interpretation and Integration Conference) [2] is a repository including two development ontology pairs (Wine and Weapons) and eight test pairs of ontologies (Animals, Sports, Computer Science, Hotels, Computer Networks, Pets, Pets (with no instances), and Russia) and describing the characteristics, basic concepts of different domains. The alignments of this benchmark tests focus on equivalent relationships and the confidence value equals to 1. Each pair of this repository is created in equivalent N3 and XML formats. The target ontologies were modified and represented in different ways.

2.5.3. OAEI BENCHMARKS 2008

The Ontology Alignment Evaluation Initiative (OAEI) is an international initiative extended from the 2004 EON Ontology Alignment Contest. This benchmark is designed with the goal for evaluating the strong and weak points of ontology

[2] http://www.atl.external.lmco.com/projects/ontology/i3con.html/

alignment methods and tools. The characteristics of this tests are that the alignments contain equivalent relations and the confidence degree is 1. Ontologies in this benchmark test are modified from the reference ontology #101 by discarding a lot of information and changing properties, using synonyms, extending structures and so on. The reference ontology #101 includes 33 classes, 24 object properties, 40 data properties, 56 named individuals and 20 anonymous individuals [3]. The benchmark consists of 111 ontologies and can be divided into three categories: 101-104 (1xx), 201-266 (2xx), and 301-304 (3xx) [17].

- Tests (1xx): this is a set of simple data tests including the reference ontology, one irrelevant ontology (focuses on wine domain) and two ontologies;
- Tests (2xx): these data tests are considered systematic tests. These data tests are created by removing or adding some information to each entity of the reference ontology, for example, instances, relations, and name concepts. Therefore, these tests allow to find out the strong points and weakness of a system as well as the response of an algorithm to the different features. The features of the entities consist of:
  - Names: the name of an entity is a string allowing to recognize entities. In these tests, names are modified in different forms, such as synonyms, abbreviation, arbitrary strings, and strings in other languages than English to create the distinguishing ontologies;
  - Comments: the comment of an entity is usually a short context to describe that entity.
  - Hierarchy: the structures represent various levels of entities in ontologies. The structures can be modified by expanding, flattening or suppressing.
  - Instances: instances are named data values of entities. They can be changed by suppressing.
  - Properties: adding some restrictions and suppressing properties are the ways to produce new versions of ontologies.
  - Classes: a class can be expanded or aggregated from some classes.
- Tests (3xx): these data tests present four real-life ontologies of bibliographic references found on the web.

[3] http://oaei.ontologymatching.org/

Analogous to the I^3CON repository, there is no single approach returning the best results in all tests.

As we shall see in next section, the classical measures for the purpose to compare the performance of matching systems are presented.

2.6. PRECISION, RECALL, F-MEASURE, AND CORRELATION COEFFICIENTS

This section reviews four classical measures for evaluation matching systems consisting of Precision, Recall, F-measure, and correlation coefficients. Before the definitions of each of these measures are presented, type of evaluation is mentioned in subsection 2.6.1.

2.6.1. TYPE OF EVALUATION

Evaluation indicates whether a system is good or not and what the strong and weak points of a system are. Depending on the purpose, evaluation is classified into three groups including competence benchmarks, comparative evaluation, and application-specific evaluation [37], which will be outlined hereafter.

- Competence Benchmarks: for this kind of evaluation, systems execute a set of well-known tasks. The results are then used for comparing and determining the quality of these systems in term of advantages, disadvantages, and the stability of systems in the certain cases.
- Comparative Evaluation: this kind of evaluation allows systems to implement a common task on a set of datasets. Moreover, the rules and the evaluation criteria should be built clearly. The quality of a system in the comparative evaluation is also determined based on run time and memory.
- Application-specific Evaluation is a kind of evaluation in which the experimental results of systems are compared to the outputs of a real application to determine the best one. This kind of evaluation does not depend on the reference alignment, which is different from the two kinds of evaluation presented above. The application-specific and comparative evaluations can be used together.

The objective of ontology matching is that ontology matching systems try to obtain as many matched pairs of entities as possible while the false results

are restricted in the amount, which are estimated by comparing to the reference alignment. Ideally, the alignment result contains all true pairs of entities and no false pair of entities. However, all systems have not fulfilled this requirement so far. For evaluating the accuracy of matching systems, the number of found correct and incorrect results needs to find out.

Fig. 2.8 illustrates resulted sets which can be obtained after the matching process.

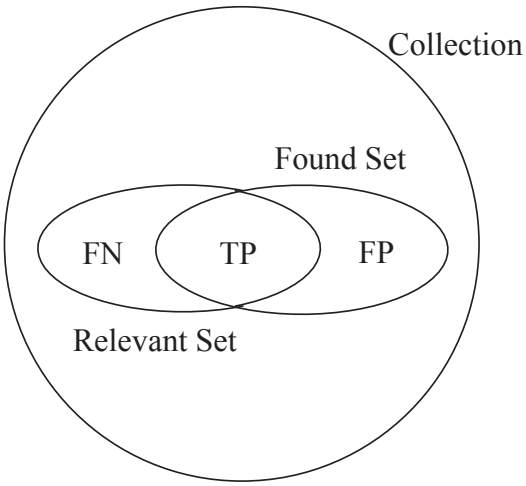

Figure 2.8: An illustrative example for resulted sets [8]

In the Fig. 2.8, TP (true positives) is the number of correct results, FP (false positives) is the number of incorrect results belonging to the rest of the found set, and FN (false negatives) is entities which belong to the rest of the relevant set. The higher the number of true positives is, the smaller the set of correct results is which are not found. The match quality measures used to compare are Precision, Recall, F-measure, and correlation coefficients. In the rest of this chapter, these classic measures are described in some more details.

2.6.2. PRECISION

Precision is defined as the number of true positives divided by the total number of found results and is expressed as follows:

$$Precision = \frac{No.\_correct\_found\_correspondences}{No.\_found\_correspondences} = \frac{TP}{TP+FP} \tag{2.1}$$

As can be seen in Eq. (2.1), Precision takes values from the interval [0, 1]. In case every found correspondence is correct, the precision is perfect and its value

equals to 1 while all relevant results might be not retrieved.

2.6.3. RECALL

Recall is defined as the number of true positives divided by the total number of existing relevant results and is expressed as the following equation.

$$Recall = \frac{No.\_correct\_found\_correspondences}{No.\_existing\_correspondences} = \frac{TP}{TP+FN} \tag{2.2}$$

As expressed in Eq. (2.2), Recall is a value in the range [0, 1]. In case all correct results are retrieved, the recall is perfect and its value equals to 1 in which the incorrect results might be also retrieved.

2.6.4. F-MEASURE

Normally, Precision and Recall are combined together to produce F-measure. F_β is used to determine the effectiveness measure and was obtained by van Rijsbergen [107] such that:

$$F_\beta = \frac{1}{\frac{\alpha}{Precision} + \frac{1-\alpha}{Recall}} \tag{2.3}$$

where

$$\alpha = \frac{1}{1+\beta^2}$$

According to Rijsbergen [107], the F_β "measures the effectiveness of retrieval with respect to a user who attaches β times as much importance to recall as precision". The traditional F-measure is the weighted harmonic mean of its Precision and Recall values (also called the F-measure or the F_1 value or the balanced F-measure because Precision and Recall values are equally weighted). Consequently, F-measure is written as follows:

$$F-measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2.4}$$

The F-measure in the Eq. (2.4) is used in this thesis. In Eq. (2.4), F-measure is a value in the range [0,1]. The F-measure value equals to 1 if only if the Precision and Recall values of 1. In case every found correspondence is incorrect, the Precision and Recall values equal to 0. As a result, the F-measure is equivalent to 0.

2.6.5. CORRELATION COEFFICIENTS

Another evaluation way is correlating. In this thesis, Pearson's correlation coefficient is used to evaluate and compare semantic similarity measures. Suppose that two datasets $(x_1, ..., x_n)$ and $(y_1, ..., y_n)$ contain n values, Pearson's correlation coefficient is a measure determining the linear relationship between these sets of data and is defined as:

$$r = \frac{\sum_{i=1}^{n} (x_i - \overline{x}) * (y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \overline{x})^2} * \sqrt{\sum_{i=1}^{n} (y_i - \overline{y})^2}} \tag{2.5}$$

where

$$\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

2.7. SUMMARY

In this chapter, issues related to Semantic Web, ontology and its components (e.g. classes, attributes, relationships, and instances), and the languages of ontologies were reviewed. In addition, techniques used in ontology matching to create the alignment were also given. Similarity functions, benchmark tests (for example, R&G and M&C, I^3CON 2004, and OAEI Benchmarks 2008), and classical measures (e.g. Precision, Recall, F-measure, and correlation coefficients) for evaluating the quality of match results were provided together, which will be used in the following chapters. Furthermore, applications of ontologies and ontology matching were also discussed.

The following chapter introduces existing techniques and ontology matching systems connecting to our approaches in this thesis.

3

Related Work

The aim of this chapter is to introduce related work in the field of ontology matching systems. From the point of view of the input information levels, the approaches of the systems can be classified into four main categories including lexical-based, structure-based, instance-based, and a combination of them. We however start with some single measures related to our approach in general. An overview about lexical methods is shown in section 3.1. A number of the existing semantic matching measures is illustrated in section 3.2 of this chapter. In the following section, several structural approaches will be discussed. Finally, the methods for finding the final alignment developed in this thesis apply different techniques, and therefore, the state of the art of available ontology matching systems will be reviewed in section 3.4, before we conclude the chapter in section 3.5.

3.1. Lexical Techniques

Lexical-based techniques are those depending on lexical chains of entities in order to decide the similarity values between these entities. The lexical similarity measures are usually used to match short strings such as entity names in ontologies, protein sequences and letter strings. In the following subsections, a brief description of these measures taken from [85] is presented.

3.1.1. Dice Coefficient

Dice coefficient (also called coincidence index) computes the similarity of two terms A and B as the ratio of two times the size of the intersection divided by the total number of samples in these sets and is given as [22]

$$sim(A,B) = \frac{2h}{a+b} \tag{3.1}$$

where A and B are different terms, h is the number of common samples in A and B, and a, b are the numbers of samples in A and B, respectively. Accordingly, the higher the number of common samples in A and B, the more their similarity increases.

Dice's measure can be described as

$$\begin{aligned} sim(A,B) &= \frac{2|A \cap B|}{|A| + |B|} \\ &= \frac{2|A \cap B|}{2|A \cap B| + |A \backslash B| + |B \backslash A|} \end{aligned} \tag{3.2}$$

3.1.2. N-grams Approach

N-grams of a sequence are all subsequences with a length equals to n. The items in these subsequences can be characters, tokens in contexts or signals in speech corpus. For example, n-grams of the string *ontology* with $n = 3$ consist of {*ont, nto, tol, olo, log, ogy*}. In case of n-grams of size 1, 2 or 3 they are also known as unigram, bigram or trigram, respectively. The n-grams approach is useful for comparing strings in which the number of common n-grams between two given strings are taken. Let $|c_1|$, $|c_2|$ are lengths of strings c_1 and c_2, respectively, the similarity between these strings can be presented as [37]

$$sim(c_1, c_2) = \frac{|ngram(c_1) \cap ngram(c_2)|}{\min(|c_1|, |c_2|) - n + 1} \tag{3.3}$$

The Eq. (3.3) can be reformulated as follows:

$$sim(c_1, c_2) = \frac{|ngram(c_1) \cap ngram(c_2)|}{\min(|ngram(c_1)|, |ngram(c_2)|)} \tag{3.4}$$

N-grams method is widely used in natural language processing, approximate matching, plagiarism detection, bioinformatics and so on. Some measures apply n-grams approach to calculate the similarity between two objects [4, 53, 65]. The combination of Dice and n-grams methods in [4, 65] to match two given concepts in ontologies is shown below.

3.1.3. KONDRAK'S AND ALGERGAWY'S METHODS

Kondrak [65] develops and uses a notion of n-grams similarity for calculating the similarities between words. In this method, the similarity can be written as

$$sim(c_1, c_2) = \frac{2|ngram(c_1) \cap ngram(c_2)|}{|ngram(c_1)| + |ngram(c_2)|} \tag{3.5}$$

As can be seen in Eq. (3.2) and Eq. (3.5), Kondrak's method is a specific case for Dice's metric in which the samples correspond to n-grams.

Matching two ontologies is presented by Algergawy et al. [4], in which three similarity methods are combined in a name matcher phase. Furthermore, Dice's expression is implemented to obtain similarities between concepts by using trigrams. Particularly, this measure applies the set of trigrams in compared strings c_1 and c_2 instead of using the number of samples in datasets. Therefore, Algergawy's method is a specific case of Kondrak's metric.

3.1.4. JACCARD SIMILARITY COEFFICIENT

Jaccard measure [54] is developed to find out the distribution of the flora in areas. The similarity related to frequency of occurrence of the flora is the number of species in common to both sets with regard to the total number of species.

Let A and B be arbitrary sets. Jaccard's metric can be normalized and is presented as [54]

$$\begin{aligned} sim(A,B) &= \frac{|A \cap B|}{|A \cup B|} \\ &= \frac{|A \cap B|}{|A \cap B| + |A \backslash B| + |B \backslash A|} \end{aligned} \tag{3.6}$$

Applying n-grams approach to Jaccard's measure leads to the following expression

$$sim(c_1, c_2) = \frac{|ngram(c_1) \cap ngram(c_2)|}{|ngram(c_1) \cup ngram(c_2)|} \tag{3.7}$$

As can be seen in equations (3.5) and (3.7), Kondrak and Jaccard measures are quite similar. Kondrak uses the total samples of two sets instead of the union of these sets as in Jaccard's equation.

3.1.5. Needleman-Wunsch Measure

The Needleman-Wunsch measure [80] is proposed to determine the similarities of the amino acids in two proteins. This measure pays attention to the maximum number of amino acids of one sequence that can be matched with another. Therefore, it is used to achieve the best alignment. A maximum score matrix $M(i,j)$ is built recursively, such that

$$M(i,j) = \max \begin{cases} M(i-1,j-1) + s(i,j) & \text{Aligned} \\ M(i-1,j) + g & \text{Deletion} \\ M(i,j-1) + g & \text{Insertion} \end{cases} \tag{3.8}$$

where $s(i,j)$ is the substitution score for characters i and j, and g is the gap score in which a gap is inserted between the characters so that similar successive characters are aligned.

3.1.6. Hamming Distance

Hamming distance [48] only applies to strings of the same sizes. With this measure, the difference between two input strings is the minimum number of substitutions that could have changed one string into the other. In case of different string lengths ($|c_1| \neq |c_2|$) and $|\{i|1 \leq i \leq \min(|c_1|,|c_2|)\}|$, Hamming distance $dis(c_1,c_2)$ is modified as [37]

$$dis(c_1,c_2) = \frac{\left(\sum_{i=1;c_1[i]\neq c_2[i]}^{\min(|c_1|,|c_2|)} 1\right) + ||c_1| - |c_2||}{\max(|c_1|,|c_2|)} \tag{3.9}$$

where $|c_1|$, $|c_2|$ are string lengths, and $c_1[i]$, $c_2[i]$ are the i^{th} characters in two strings c_1 and c_2, respectively.

Besides using only the operation of substitutions, the Levenshtein distance applying insertions or deletions for comparing strings of different lengths is presented in the succeeding section.

3.1.7. Levenshtein Distance

The Levenshtein distance (also called Edit distance) [68] is a well-know string metric calculating the amount of differences between two given strings and then returning a value. This value is the total cost of the minimum number of operations needed to transform one string into another. Three types of operations are used including the substitution of a character of the first string by a character of

the second string, the deletion or the insertion of a character of one string into other. The total cost of the used operations is equal to the sum of the costs of each of the operations.

Let c_1 and c_2 be two arbitrary strings. The similarity measure for two strings $sim(c_1, c_2)$ is described as [73]

$$sim(c_1, c_2) = \max\left(0, \frac{\min(|c_1|, |c_2|) - ed(c_1, c_2)}{\min(|c_1|, |c_2|)}\right) \tag{3.10}$$

where $|c_1|$, $|c_2|$ are lengths of strings c_1 and c_2, respectively, and $ed(c_1, c_2)$ is Levenshtein measure. Note that the cost assigned to each operation here equals to 1. In case edit distance of these strings is greater the minimum number of characters of the lengths of two strings, the formula $(\min(|c_1|, |c_2|) - ed(c_1, c_2))$ becomes negative. In this case, the similarity value between these strings is assigned to 0.

3.1.8. JARO-WINKLER MEASURE

The Jaro-Winkler measure [141] is based on the Jaro distance metric [56] to compute the similarity between two strings. The Jaro-Winkler measure $sim(c_1, c_2)$ between c_1 and c_2 strings can be defined as follows:

$$sim(c_1, c_2) = sim_{Jaro}(c_1, c_2) + ip(1 - sim_{Jaro}(c_1, c_2)) \tag{3.11}$$

where i is the number of the first common characters (also known as the length of the common prefix), p is a constant and is assigned to 0.1 in Winkler's work [141] and $sim_{Jaro}(c_1, c_2)$ is the Jaro metric, defined as

$$sim_{Jaro}(c_1, c_2) = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|c_1|} + \frac{m}{|c_2|} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases} \tag{3.12}$$

In Eq. (3.12), m is the number of matching characters and t is the minimum number of the interchanges of characters to different positions in one string such that the matching characters in two strings are in the same order.

3.1.9. TVERSKY'S MODEL

In Tversky's ratio model [133], determination of the similarity among objects is related to features of these objects. In particular, the similarity value of object o_1 to object o_2 depends on their shared and different features, so that

$$sim(o_1, o_2) = \frac{f(\phi(o_1) \cap \phi(o_2))}{f(\phi(o_1) \cap \phi(o_2)) + \beta f(\phi(o_1) \backslash \phi(o_2)) + \gamma f(\phi(o_2) \backslash \phi(o_1))} \tag{3.13}$$

where $\phi(o_1)$ and $\phi(o_2)$ represent features of o_1 and o_2, respectively, f is a function of a set of features, $\phi(o_1) \cap \phi(o_2)$ presents common features of both o_1 and o_2, $\phi(o_i) \backslash \phi(o_j)$ describes features being held by o_i but not in o_j, $(i, j = 1,2)$. The parameters β and γ are adjusted and depend on which features are taken into account. Therefore, in general this model is asymmetric, it means, $sim(o_1, o_2) \neq sim(o_2, o_1)$. This model is also a general approach applied in many matching functions in the literature as well as domains [99, 114].

3.2. SEMANTIC TECHNIQUES

The semantic-based techniques have been proposed to achieve semantic similarity and relatedness values between two considered concepts. In this section, an introduction to similarity measures extended from [84] is shown.

3.2.1. RADA'S APPROACH

Rada et al. [103] proposed a semantic similarity measure based on the edge-based approach. In this measure, they compute the conceptual distance between two concepts based on hypernym/hyponym relations in the taxonomy. The conceptual distance is defined as the shortest path length over all pairwise combinations of nodes. Although this measure is simple, it is only implemented in the specific domain such as medical domain.

3.2.2. LEACOCK&CHODOROW MEASURE

The principle of similarity computation of Leacock and Chodorow measure [67] is also based on edge counting method. The authors calculate semantic similarity measure by considering the length of the shortest path that connects two concepts C_1, C_2 and the maximum depth in WordNet hierarchy. The measure is described as follows:

$$sim_{LC}(C_1, C_2) = -\log \frac{len(C_1, C_2)}{2D} \tag{3.14}$$

where $len(C_1, C_2)$ represents the length of the shortest path between two concepts C_1 and C_2, D represents the maximum depth in WordNet.

3.2.3. SUSSNA'S MEASURE

Sussna et al.'s work [127] deals with the basis of edge counting-based measures. However, their measure is extended with weighted edges in which weights depend on relation types. The conceptual distance is a function of the shortest weighted path between two nodes and is expressed as follows:

$$dist_S(C_1, C_2) = \frac{w(C_1 \rightarrow_r C_2) + w(C_2 \rightarrow_{r'} C_1)}{2 * \max(depth(C_1), depth(C_2))} \tag{3.15}$$

where $\rightarrow_r$ and $\rightarrow_{r'}$ are relationships of type r and its inverse (for example, "is-a" and "is-a-subclass-of" are inverse relationships), $w(C_1 \rightarrow_r C_2)$, $w(C_2 \rightarrow_{r'} C_1)$ are the weights of edges of types r and r', and $depth(C_1)$, $depth(C_2)$ are the depths of nodes, respectively.

3.2.4. RESNIK'S METRIC

Resnik [105] applies an information content approach to obtain semantic similarity of two words in an "is-a" taxonomy. General property of information content approach is that the more abstract a concept, the lower its information content because information content of a concept depends on the annotation statistics related to that concept. Resnik's metric can be used for any domain, but requires a corpus to calculate the frequencies of concept occurrences. The key idea of Resnik's approach is that the information content of the nearest common ancestor of two concepts represents the similarity value of these two nodes:

$$sim_R(C_1, C_2) = IC(nca(C_1, C_2)) \tag{3.16}$$

where $nca(C_1, C_2)$ is the nearest common ancestor of two nodes and IC means the information content. However, one disadvantage of this approach is that the similarities of pairs of any two concepts are the same in case the nearest common ancestor of these pairs of concepts is the same. A number of measures such as Jiang&Conrath [59] and Lin et al. [71] measures later extend Resnik's measure with more factors.

3.2.5. JIANG&CONRATH'S AND LIN'S METRICS

While Resnik measure uses the information content of the nearest common ancestor of concepts, Jiang&Conrath [59] and Lin et al. [71] measures combine the information content of individual concepts and the nearest common ancestor of

these concepts together. The distance function of Jiang&Conrath and Lin's similarity measure can be defined as follows, respectively.

$$dist_{JC}(C_1, C_2) = IC(C_1) + IC(C_2) - 2 * IC(nca(C_1, C_2)) \tag{3.17}$$

and

$$sim_L(C_1, C_2) = \frac{2 * IC(nca(C_1, C_2))}{IC(C_1) + IC(C_2)} \tag{3.18}$$

3.2.6. Alvarez's and Li2003's Measures

For Alvarez's and Li2003's measures, the authors combine three factors, that are the shortest path length, the gloss overlapping and the depth of the nearest common ancestor of nodes C_1 and C_2 to get the similarity score of these nodes (see more in detail [5, 70]).

3.2.7. Bin's Measure

Shi Bin et al.'s measure computes the path lengths, local density and the connection power between two nodes and then integrates them with edge weights (see more in detail [13]).

3.2.8. Wu&Palmer's Metric

Wu and Palmer [142] proposed a similarity metric based on the edge-based approach. In their measure, the similarity is determined by the depths of the two input concepts in the taxonomy along with the depth of the nearest common ancestor. Particularly, it is expressed by the following formula:

$$sim_{WP}(C_1, C_2) = \frac{2 * N}{N_1 + N_2} \tag{3.19}$$

where N_1, N_2, and N are the lengths of the paths which separate nodes C_1, C_2, and the nearest common ancestor of these nodes from the root, respectively. The feature of this method is that it is based on edge-counting so it is computationally efficient compared to other approaches. For this reason, Wu&Palmer's method is considered as a backbone for our proposed measure. However, the similarity values obtained by this method are slightly far from the human judgements because the authors ignore the semantic relationships of concepts which should be considered.

3.2.9. SLIMANI'S MEASURE

In [120] the authors improve Wu&Palmer's measure by multiplying a penalization factor in order to deal with inadequate situations where the similarity of two neighbors should be lower than the similarity of two concepts in the same hierarchy. The Slimani's measure is defined as

$$sim_{Slimani}(C_1, C_2) = \frac{2 * N}{N_1 + N_2} * PF(C_1, C_2) \tag{3.20}$$

where $PF(C_1, C_2)$ is a penalization factor such that

$$PF(C_1, C_2) = (1 - \lambda) * (\min(N_1, N_2) - N) + \lambda * (|N_1 - N_2| + 1)^{-1} \tag{3.21}$$

and N_1, N_2, and N are the lengths of the paths from each node to the root and from the nearest common ancestor of two nodes to the root, respectively, the coefficient λ indicates 0 in case two concepts in the same hierarchy or 1 in case two concepts in neighborhood, and $\min(N_1, N_2)$ represents the minimum value between C_1 and C_2.

In summary, all measures aforementioned have a common point which depends on the structure such as "is-a" relations in WordNet hierarchy and on additional information. However, a major problem with these measures is that they omit relationships connecting the input concepts. Since WordNet is a complex network, there are a lot of concepts connecting to others in irregular way i.e., a concept might have many fathers or the relationships are overlapping. Furthermore, WordNet is a semantic network so we argue that the different relations between concepts should be used to compute the semantic similarity values. Our measure will pay attention to this feature (see more in detail chapter 5).

3.3. STRUCTURAL TECHNIQUES

The structure-based techniques consider the position of the concepts and the relationships among ontologies. The intuition of these approaches is that two concepts of the ontologies are similar if their structures are similar. However, structure-based methods usually calculate the similarity of two concepts in the ontologies based on parents, neighbors, children or leaves. Many ontology matching methods have been proposed to obtain alignments among entities of given ontologies. In this section, some approaches related to structural techniques are briefly presented.

3.3.1. Inexact Matching Approach

In the paper [29], the authors consider the matching of two ontologies as a maximum likelihood problem and resolve it by using the expectation-maximization technique. In particular, ontology schemas are modeled as directed graphs and then the structural, lexical, and instance methods are applied to take mappings between these graphs. In this approach, the structural similarities between nodes take account of neighbors. Moreover, in case choosing exact matched results is difficult, inexact matching is employed to find a best possible correspondence.

3.3.2. OLA Tool

OLA [38] is an automatic matching tool which constructs graphs from the input ontologies and combines two techniques, that are concept-based and structure-based techniques, to match these graphs. However, in the alignment process, OLA only considers contributions of all the similarities of neighbors in the same type.

3.3.3. ASMOV Algorithm

The approach used in ASMOV [58] integrates lexical, structural and extensional methods for the calculation of similarities and then performs semantic verification based on the pre-alignment in which the alignment is obtained by the greedy algorithm. For calculating the relational similarities, the parents and children of the entities are taken into account. Algorithm is similar to OLA, but it has more flexibility to the calculations for different features.

3.3.4. RiMOM System

To achieve the optimal alignment, RiMOM [129] applies different strategies such as lexical-based, constraint-based and linguistic-based techniques. Additional, it also uses a taxonomic technique. However, in this approach, it only considers the direct super-concepts and sub-concepts of a node. The individual results are then combined by using the linear-interpolation method.

3.3.5. VBOM Technique

VBOM [34] is a structural-based technique for automated ontology alignment which matches entities based on vector similarity algorithms and then applies two heuristic rules to enhance the matching quality. This approach transfers the

relationships and entities of the given ontologies into the space of vectors in N dimensions. Particularly, each entity contains a vector with the weights of its ancestors and successors classes. The similarity degree between two concepts is computed by the cosine of the pair of these concept vectors.

3.3.6. MLMA+ APPROACH

MLMA+ algorithm [3] and its improvement [2] are approaches using neighbor searching techniques to find the best method for matching. These approaches consider the given ontologies as labeled directed graphs, apply well-known individual matching techniques at two levels and then combine the techniques to improve the overall resulting correspondences.

3.3.7. DSI METHOD

In [126] the structural similarities of nodes are calculated by the similarities of pairs of parents as well as these of siblings. The authors proposed the DSI method which allows ancestors of a concept to identify that concept. It is known that different ontologies have different characteristics. In case two ontologies do not have the same depth they can omit some pairs of nodes similarities because they only take care of pairs of ancestors at the same generation.

3.3.8. ANCHOR-PROMPT ALGORITHM

The Anchor-PROMPT is an algorithm that calculates semantically similarities between nodes in graphs. In [90] the authors analyze similar paths between a set of anchor matches to obtain new concept mappings. In other words, if two pairs of nodes from the input ontologies are similar and there are paths connecting the nodes then the entities in those paths are similar. This measure tries to find relationships between entities based on the primary relations recognized before.

3.3.9. SIMILARITY FLOODING ALGORITHM

Similarity flooding algorithm is established in [74]. The authors build PCGs and use structural characteristics for propagating similarities between elements on a directed labeled graph. This approach finds similarities in the graph structure. It spreads the similarity from similar nodes to their neighbors and back based on propagation coefficients.

3.3.10. The Structure-based Similarity Spreading Method

The structure-based similarity spreading method [139] executes ontology matching in three phases. In the first phase, the centroid concepts are selected from the input ontologies by using linguistic similarities between entities. Note that only concepts in two ontologies in which their similarity values equal to 1 are picked. In the next step, the local similarities of nodes are established by the centroid concepts through a partition technique. The structural method based on the similarity flooding algorithm is then used to improve the similarities and the greedy matching method is employed to determine the best match solution.

3.3.11. Other Approaches

The author in [118] represents ontologies and schemas as graphs. Using these graphs can help to calculate the weighted value for each node on the graph using the lexical similarity of ancestors. However, this method only calculates the values up to the grandparent level.

The authors in [24, 72] map structural similarities based on children and leaves. The COMA approach [24] assumes that two non-leaf elements can be considered similar if their children and leaves are similar while the Cupid approach [72] realizes that two objects which are not leaves are similar if their leaf sets are highly similar.

The proposed method in [1] implements three phases to retrieve the possible matching solution. The lexical and structural measures are applied in the first two phases in which the lexical similarity method produces a bag of words and the structural approach generates a grid around each entity in the ontologies by linking it with its neighbors and the neighbor of its neighbors. The structural similarity can be computed based on these grids. The third phase then combines the component matrices to achieve the final results.

3.4. Ontology Matching Systems

In the previous chapter, we have pointed out that ontology matching systems can be produced by combining several different techniques. Many of the systems discussed below are covered in the book by Jérôme and Euzenat [37]. In the scope of this thesis, some of the most systems that have been applied so far to the task of matching based on structures are summarized. In general, structural-based ontology matching systems consider information of structure in the hierarchy

to find the matching entities of given ontologies, in which our approach is also concentrated on. Another reason to chose these systems is that these systems are evaluated based on the same benchmark, the OAEI 2008 test set, which is convenient and fair in comparison.

3.4.1. CIDER

CIDER [44] applies ontology matching techniques to determine similarities between classes and properties based on the labels, structures, instances, and semantics in OWL or RDF ontologies. This system extracts terms based on their semantic by using an external resources such as WordNet up to a fixed depth. The similarities between these terms are then computed based on lexical, taxonomical and relational techniques. In particular, the system employs Levenshtein edit distance metric for calculating similarities between labels and descriptions, a vector space model to achieve structural similarities, and an artificial neural network to integrate similarities. CIDER uses thresholds to extract one-to-one alignments.

3.4.2. SPIDER

Spider [112] combines two subsystems: CIDER and Scarlet where Scarlet investigates online ontologies automatically to obtain different types of relations between two concepts, for example, equivalence, subsumption, disjointness, and named relationships by applying derivation rules.

3.4.3. GEROMESUITE

GeRoMeSuite [61] is a flexible model management tool using the metamodel GeRoMe [60]. This system executes a number of matching techniques, for example, string-based, semantics-based, and structure-based methods. Additionally, GeRoMeSuite approach can load XML Schema and OWL ontologies and then performs alignment task.

3.4.4. MLMA+

MLMA+ [3] implements a matching algorithm in two levels where the structure-based method at the second level is followed by the name and linguistic similarities at the first level to obtain the final matching results. Besides, MLMA+ suggests a list of similarity measures which should be used to improve the overall

similarity results. The final alignment of this system is a many-to-many cardinality.

3.4.5. Anchor-Flood

Similar to the MLMA+ system, Anchor-Flood [116] combines lexical-based, structure-based, and semantics-based similarity measures to calculate the correspondences between fragments in RDFS and OWL ontologies and then returns one-to-one alignments. However, this approach computes the similarity between terms through the Winkler-based string metric, which is different from MLMA+.

3.4.6. DSSim

DSSim [79] is an ontology matching framework using the structures in the hierarchy to find the confidence degrees between concepts and properties in the two large scale ontologies. In addition, the Monge-Elkan and Jaccard similarity measures are used for calculating similarities between strings and WordNet dictionary, which can be employed in determining semantics. DSSim system utilizes inputs as OWL and SKOS ontologies and gives outputs as one-to-one alignments.

3.4.7. Lily

Lily [137] combines three ontology matchers including Generic ontology matching method (GOM), Large scale ontology matching (LOM), and Semantic ontology matching (SOM) to compute one-to-one alignments. After a preprocessing step, Lily applies measures to determine the similarity between entities in given ontologies including string-based, structure-based, semantics-based, and instance-based comparison algorithms. Then ontology mapping debugging technique is applied for the post-processing step to find the best possible matching solution.

3.4.8. MapPSO

MapPSO [14] combines the SMOA string distance, structure-based, WordNet-based and vector space similarity approaches, and ordered weighted average method to obtain one-to-one matching between concepts and properties in large OWL ontologies. In addition, the MapPSO approach considers the finding of the correspondences as an optimization problem.

3.4.9. TAXOMAP

TaxoMap [46] further develops its previous version presented in [147]. In this new implementation, TaxoMap applies ontology matching techniques including the linguistic, 3-grams, structural similarity methods, and heuristic rules to obtain one-to-many cardinality between concepts. Besides, TaxoMap approach only concentrates on the labels and the relationships between the concepts in the hierarchy. The difference from the old version is that TaxoMap system runs on large scale ontologies.

3.4.10. AKBARI&FATHIAN

Akbari&Fathian [1] is a combined approach to identify correspondences between entities in the source and target ontologies. This system computes the lexical similarities of class names, object properties and data properties, and the structural similarities of class names and then integrates similarity matrices to produce the final alignment by using the weighted mean.

3.4.11. AGREEMENTMAKER

AgreementMaker system [18] matches concepts in the given ontologies by comparing their information available, for example, labels, comments, annotations, and instances. This system can deal with XML, RDFS, OWL, and N3 ontologies and then applies lexical, syntactic, structural, and semantic methods. The total values are aggregated through the weighted average method to match one entity to one entity.

3.4.12. ASCO

ASCO [7] is an automatic ontology matching system. It uses RDF(S) ontologies and implements the linguistic and structural phases for finding the corresponding matches between entities in the considered ontologies. Besides, this approach applies several well-known measures, for example, Jaro-Winkler, Levenshtein, Monger-Elkan, and computes the semantic similarities based on WordNet dictionary. The weighted sum method is then used in integrating the partial similarities to yield one-to-one or one-to-many alignments. ASCO2 [6] is developed to work with OWL ontologies.

3.5. Summary

The current chapter was closed by a list of well-known ontology matching systems. We revised approaches to existing automatic structural ontology matching systems tested on the benchmark dataset of the 2008 OAEI. In addition to that, the presented overview sections provided a wide variety of related approaches. These techniques are used for almost all matching systems and will be also applied in our integrated approach for measuring the similarity values.

In the following chapters 4, 5, 6, and 7, our own contribution to the individual techniques as well as an approach combining these techniques together will be discussed.

4

Lexical Similarity Measure based on Combining Information-Theoretic and Edit Distance Measures

Measurement of similarity plays an important role in data mining and information retrieval. Several techniques for calculating the similarities between objects have been proposed so far, for example, lexical-based, structure-based and instance-based measures. In the scope of this chapter, a lexical similarity approach combining information-theoretic model and edit distance is developed to determine correspondences among the concept labels. Precision, Recall and F-measure as well as partial OAEI 2008 benchmark tests are used to evaluate the proposed method. The results show that our approach is flexible and has some prominent features compared to other lexical-based methods.

The remainder of this chapter is organized as follows. First of all, a short introduction to our measure is given in section 4.1. In section 4.2, a similarity measure taking into account text strings is proposed. In section 4.3, we describe our experimental results, give an evaluation as well as a discussion of our measure and compare it with other approaches applying Precision, Recall and F-measure. Then conclusions and future works are presented in section 4.4. Finally, a summary in section 4.5 concludes this chapter.

4.1. INTRODUCTION

A number of similarity measures for determining the similarities between objects have been proposed so far, applied in many well-known areas. Among these, lexical similarity metrics find correspondences between given strings. These measures are usually applied for ontology matching systems, information integration, bioinformatics, plagiarism detection, pattern recognition and spell checkers. The lexical techniques are based on the fact that the more the characters in strings are similar, the more the similarity values increase. Existing lexical-based measures usually based on either n-grams or Dice's approaches to obtain the similarity degrees between strings (see more section 3.1). The advantage of these measures is a good performance. Moreover, n-grams metrics could be extended in case the parameter n is adjusted. However, they have the disadvantage that they do not return reasonable results in some situations where strings are quite similar or the sets of characters are the same but their positions are different in strings. To deal with this problem, a similarity approach based on the combination of features-based and element-based measures is proposed. In particular, it is combined from information-theoretic model and edit-distance measure. Consequently, common and different properties with respect to characters in strings as well as editing and non-editing operations are considered.

4.2. COMBINING INFORMATION-THEORETIC AND EDIT DISTANCE MEASURES

4.2.1. OUR LEXICAL SIMILARITY MEASURE

In this section, a lexical similarity measure is proposed. Our approach is motivated on Tversky's set-theoretical model [133] and Levenshtein measure [68]. We agree that the similarities among entities depend on their commonalities and differences based on the intuitions in [71]. The well-known metrics applying Tversky's model take into account features of compared objects such as intrinsic information content [99, 100], the number of shared superconcepts [10], the number of common attributes, instances and relational classes [138] in ontologies. In contrast with existing approaches, the objective of our metric is to focus on the features in terms of the contents of the characters and their positions in strings. In particularly, our measure is related to editing and non-editing operations.

As mentioned earlier, Tversky's model is a general approach considering the common and different features of objects in which the different features are represented by their proportions through parameters β and γ. In our method, f reflects the cardinality of a set, a parameter α is added to the common feature in Eq. 3.13 in chapter 3. Consequently, the similarity is given by

$$sim(c_1, c_2) = \frac{\alpha|\phi(c_1) \cap \phi(c_2)|}{\alpha|\phi(c_1) \cap \phi(c_2)| + \beta|\phi(c_1) \setminus \phi(c_2)| + \gamma|\phi(c_2) \setminus \phi(c_1)|} \tag{4.1}$$

where the parameters α, β and γ are subjected to a constraint: $\alpha + \beta + \gamma = 1$.

According to Tversky's model, the similarity between two objects does not need to satisfy the symmetrical property because it depends on the remarkable feature of each object. However, regarding to our point of view the similarity of two strings should be a symmetric function, the differences between these strings have the same contribution, and the parameters β and γ can be considered to be equal. Therefore, our measure $Lex\_sim(c_1, c_2)$ can be rewritten as

$$Lex\_sim(c_1, c_2) = \frac{\alpha|\phi(c_1) \cap \phi(c_2)|}{\alpha|\phi(c_1) \cap \phi(c_2)| + \beta|\phi(c_1) \setminus \phi(c_2)| + \beta|\phi(c_2) \setminus \phi(c_1)|} \tag{4.2}$$

where $\alpha + 2\beta = 1$ and $\alpha, \beta \neq 0$.

In case $\alpha = \beta = \gamma = \frac{1}{3}$, our measure can be written as

$$Lex\_sim(c_1, c_2) = \frac{\alpha|\phi(c_1) \cap \phi(c_2)|}{\alpha(|\phi(c_1) \cap \phi(c_2)| + |\phi(c_1) \setminus \phi(c_2)| + |\phi(c_2) \setminus \phi(c_1)|)} \tag{4.3}$$

$$= \frac{|\phi(c_1) \cap \phi(c_2)|}{|\phi(c_1) \cup \phi(c_2)|} \tag{4.4}$$

which coincides with the Jaccard's measure.

The representation of the Dice's approach can be obtained by setting $\beta = \gamma = \frac{1}{2}\alpha$. Indeed,

$$Lex\_sim(c_1, c_2) = \frac{\alpha|\phi(c_1) \cap \phi(c_2)|}{\alpha|\phi(c_1) \cap \phi(c_2)| + \frac{1}{2}\alpha|\phi(c_1) \setminus \phi(c_2)| + \frac{1}{2}\alpha|\phi(c_2) \setminus \phi(c_1)|} \tag{4.5}$$

$$= \frac{2|\phi(c_1) \cap \phi(c_2)|}{|\phi(c_1)| + |\phi(c_2)|} \tag{4.6}$$

In this work, features of strings are chosen as the contents and positions of characters. It is the number of deletions, insertions and substitutions. Moreover, the Levenshtein measure is used to achieve common and different values

between two strings. The editing operations can be regarded as the difference, while non-editing can be reflected on commonalities. These values are then applied to Tversky's model.

Accordingly, common features between two strings are obtained by subtracting the total cost of the operations needed to transform one string into another from the maximum length of these strings and is represented as

$$|\phi(c_1) \cap \phi(c_2)| = \max(|c_1|, |c_2|) - ed(c_1, c_2) \tag{4.7}$$

The differences between two strings are:

$$|\phi(c_1) \backslash \phi(c_2)| = |c_1| - \max(|c_1|, |c_2|) + ed(c_1, c_2) \tag{4.8}$$

and

$$|\phi(c_2) \backslash \phi(c_1)| = |c_2| - \max(|c_1|, |c_2|) + ed(c_1, c_2) \tag{4.9}$$

respectively.

Our similarity measure for two strings (c_1, c_2) based on Levenshtein measure becomes:

$$\begin{aligned} & Lex\_sim(c_1, c_2) = \\ = \; & \frac{\alpha(\max(|c_1|, |c_2|) - ed(c_1, c_2))}{\alpha(\max(|c_1|, |c_2|) - ed(c_1, c_2)) + \beta(|c_1| + |c_2| - 2\max(|c_1|, |c_2|) + 2ed(c_1, c_2))} \end{aligned} \tag{4.10}$$

where $|c_1|$, $|c_2|$ are lengths of strings c_1 and c_2, respectively; $ed(c_1, c_2)$ is Levenshtein measure and $\alpha + 2\beta = 1$.

In case $\beta = \frac{1}{2}\alpha$ and the lengths of two strings are the same then our measure can be formalized as:

$$\begin{aligned} & Lex\_sim(c_1, c_2) = \\ = \; & \frac{\alpha(\max(|c_1|, |c_2|) - ed(c_1, c_2))}{\alpha(\max(|c_1|, |c_2|) - ed(c_1, c_2)) + \beta(|c_1| + |c_2| - 2\max(|c_1|, |c_2|) + 2ed(c_1, c_2))} \\ = \; & \frac{2\alpha(\max(|c_1|, |c_2|) - ed(c_1, c_2))}{\alpha(|c_1| + |c_2|)} \end{aligned} \tag{4.11}$$

When the lengths of two strings are the same, we have $\max(|c_1|, |c_2|) = \min(|c_1|, |c_2|) = |c_1| = |c_2|$, substitution in Eq. (4.11) yields

$$\begin{aligned} Lex\_sim(c_1, c_2) &= \frac{2\alpha(\max(|c_1|, |c_2|) - ed(c_1, c_2))}{\alpha(|c_1| + |c_2|)} \\ &= \frac{2\alpha(\min(|c_1|, |c_2|) - ed(c_1, c_2))}{2\alpha(\min(|c_1|, |c_2|))} \\ &= \frac{\min(|c_1|, |c_2|) - ed(c_1, c_2)}{\min(|c_1|, |c_2|)} \end{aligned} \tag{4.12}$$

which is similar to the Levenshtein's measure.

4.2.2. PROPERTIES OF OUR LEXICAL SIMILARITY MEASURE

In this section, the properties of our similarity measure Lex_sim are discussed. Our measure represented in Eq. (4.10) satisfies three properties of a similarity measure (see Section 2.4.1) as follows [37]:

- Positiveness: $\forall c_1, c_2 : Lex\_sim(c_1, c_2) \geq 0$

  Proof. Without loss of generality, we can assume that $|c_1| \geq |c_2|$.
  Therefore, we have $|c_1| - |c_2| \leq ed(c_1, c_2) \leq |c_1|$.
  $\Rightarrow 0 \leq 2|c_2| - 2|c_1| + 2ed(c_1, c_2) \leq |c_1| + |c_2| - 2|c_1| + 2ed(c_1, c_2)$
  and $ed(c_1, c_2) \leq |c_1|$
  $\Rightarrow 0 \leq |c_1| + |c_2| - 2\max(|c_1|, |c_2|) + 2ed(c_1, c_2)$
  and $\max(|c_1|, |c_2|) - ed(c_1, c_2) \geq 0$
  Because $\alpha, \beta > 0$, $Lex\_sim(c_1, c_2) \geq 0$. □

- Maximality: $\forall c_1, c_2, c_3 : Lex\_sim(c_1, c_1) \geq Lex\_sim(c_2, c_3)$

  Proof. The values of our measure were taken in the range of [0, 1]. Indeed, $|c_2| + |c_3| - 2\max(|c_2|, |c_3|) + 2ed(c_2, c_3) \geq 0$ and $\max(|c_2|, |c_3|) - ed(c_2, c_3) \geq 0$, so $Lex\_sim(c_2, c_3) \leq 1$.
  Moreover, we have $Lex\_sim(c_1, c_1) = 1$.
  Therefore, $Lex\_sim(c_1, c_1) \geq Lex\_sim(c_2, c_3)$.
  □

  $Lex\_sim(c_2, c_3) = 1$ if and only if $(|c_2| + |c_3| - 2\max(|c_2|, |c_3|) + 2ed(c_2, c_3)) = 0$, it means c_2 and c_3 are similar.

- Symmetry: $\forall c_1, c_2 : Lex\_sim(c_1, c_2) = Lex\_sim(c_2, c_1)$

  Proof. Because the entities c_1 and c_2 have the same contribution in our lexical measure, $Lex\_sim(c_1, c_2) = Lex\_sim(c_2, c_1)$ $\forall c_1, c_2$. □

In order to evaluate the performance of our lexical similarity measure, experiments and results are shown in the following section.

4.3. EXPERIMENTS AND DISCUSSIONS

We used ontologies taken from the OAEI benchmark 2008 to test and evaluate the performance of our measure and other approaches through comparing between their output and reference alignments. This benchmark consists of ontologies modified from the reference ontology 101 by changing properties, using synonyms, extending structures and so on. Since the measures here concentrate on calculating the string-based similarity, only ontologies relating to modified labels and the real bibliographic ontologies are chosen to evaluate. Consequently, the considered ontologies consist of 101, 204, 301, 302, 303 and 304. Actually, these chosen ontologies are quite suitable for the validation and comparison among Needleman-Wunsch, Jaro-Winkler, Levenshtein, normalized Kondrak's method combining Dice and n-grams approaches, with using the same classical metrics. These classical measures (Precision, Recall, and F-measure) are described in section 2.6.

Precision, Recall, F-measure and their average values for six pairs of ontologies are presented in Table 4.1. Note that these results in Table 4.1 are obtained by means of thresholds changed for nine different values from 0.5 to 0.9 with the increment of 0.05; in addition, two parameters including $\alpha = 0.2$ and $\beta = 0.4$ were applied. Based on each threshold value, the alignments are achieved for five participants. Then average Precision, Recall and F-measure for all these thresholds were calculated.

In Table 4.1, our measure gives premier value of average F-measure compared to those of other methods. It clearly indicates that our approach is more effective than the others. Moreover, both our measure and Levenshtein's are slightly better than Kondrak's metric for each pair of ontologies. For the ontology 101, when compared to itself, all methods above produce the values of Precision, Recall and F-measure to be 1.0. The value of Recall is quite important because it lets us estimate the number of true positives which is compared to the number of existing correspondences in the reference alignment. In general, with the same value of Recall, the measure which is better provides higher Precision. Although Recall values of Levenshtein, Kondrak, Jaro-Winkler, Needleman-Wunsch measures and ours are similar for ontology 301, our measure gives better Precision values than those of these measures. That means our approach is better than existing methods. Since ontology 301 consists of concepts which are slightly or completely modified from the reference ontology, the number of obtained true positive concepts are the same for string-based metrics mentioned before. Thus, in this case Recall measures have the same values in all methods. Because on-

Table 4.1: Average Precision, Recall and F-measure values of different methods for six pairs of ontologies with thresholds changed (Pre.=Precision, Rec.=Recall, F.=F-measure).

Measures		101	204	301	302	303	304	Avg.
Levenshtein	Pre.	1.0	0.982	0.835	0.929	0.880	0.955	0.930
	Rec.	1.0	0.889	0.591	0.435	0.784	0.930	0.771
	F.	1.0	0.933	0.692	0.592	0.829	0.942	0.832
Jaro-Winkler	Pre.	1.0	0.969	0.604	0.595	0.563	0.906	0.773
	Rec.	1.0	0.956	0.591	0.469	0.833	0.933	0.797
	F.	1.0	0.963	0.598	0.524	0.672	0.919	0.779
Needleman-Wunsch	Pre.	1.0	0.933	0.606	0.659	0.618	0.899	0.786
	Rec.	1.0	0.909	0.591	0.459	0.778	0.930	0.778
	F.	1.0	0.921	0.598	0.541	0.688	0.914	0.777
Kondrak	Pre.	1.0	0.967	0.797	0.871	0.810	0.951	0.899
	Rec.	1.0	0.774	0.591	0.435	0.772	0.933	0.751
	F.	1.0	0.860	0.679	0.580	0.790	0.942	0.809
Our measure	Pre.	1.0	0.989	0.888	0.949	0.952	0.965	0.957
	Rec.	1.0	0.842	0.591	0.435	0.778	0.926	0.762
	F.	1.0	0.910	0.710	0.596	0.856	0.945	0.836

tology 204 only contains concepts modified from the reference one by adding underscores, abbreviations and so on, the measures achieve the rather high results of F-measure. Ontology 304 has similar vocabularies to the ontology 101, so Precision and Recall values which are achieved for this pair of ontologies are also good. Jaro-Winkler measure is also known as a good approach because its average Recall value is slightly higher than others. However its average Precision is significantly lower than others, for example: 0.773 compared to 0.930, 0.786, 0.899 and 0.957. Therefore, the number of obtained false positive concepts of Jaro-Winkler is higher than other measures. This phenomenon occurs in the same manner in the pairs of ontologies 302 and 303.

Table 4.2: Average Precision, Recall and F-measure values of different methods for six pairs of ontologies with nine thresholds (Pre.=Precision, Rec.=Recall, F.=F-measure).

Thresholds		0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9
Levenshtein	Pre.	0.8942	0.8921	0.8921	0.8975	0.9404	0.9458	0.9458	0.9823	0.9821
	Rec.	0.7886	0.7743	0.7743	0.7743	0.7743	0.7743	0.7693	0.7642	0.7486
	F.	0.8255	0.8173	0.8173	0.8199	0.8377	0.8402	0.8373	0.8451	0.8359
Jaro-Winkler	Pre.	0.6755	0.6799	0.6814	0.7003	0.7638	0.8273	0.8438	0.8914	0.8914
	Rec.	0.8082	0.8082	0.8009	0.8009	0.8009	0.7886	0.7886	0.7886	0.7886
	F.	0.7268	0.7298	0.7277	0.7402	0.7761	0.7987	0.8074	0.8244	0.8244
Needleman-Wunsch	Pre.	0.6346	0.6357	0.6531	0.7485	0.8281	0.8544	0.8674	0.9049	0.9456
	Rec.	0.7939	0.7866	0.7866	0.7866	0.7794	0.7743	0.7743	0.7693	0.7486
	F.	0.6929	0.6916	0.7031	0.7607	0.795	0.8024	0.8093	0.8208	0.8253
Kondrak	Pre.	0.8511	0.851	0.8593	0.8593	0.9049	0.912	0.9374	0.937	0.9823
	Rec.	0.7836	0.7785	0.7693	0.7693	0.7592	0.7491	0.7196	0.7146	0.7146
	F.	0.8076	0.8048	0.802	0.802	0.8141	0.8115	0.8027	0.7989	0.8118
Our measure	Pre.	0.9147	0.9404	0.9404	0.9458	0.9458	0.9823	0.9823	0.9821	0.9821
	Rec.	0.7836	0.7743	0.7743	0.7743	0.7693	0.7693	0.7642	0.7385	0.709
	F.	0.8321	0.8377	0.8377	0.8402	0.8373	0.848	0.8451	0.8294	0.8088

Let us consider average Precision, Recall and F-measure values of different methods for six pairs of ontologies with nine thresholds separately. As can be seen in Table 4.2, our measure and Levenshtein measure are still better than Kondrak's measure in general. For thresholds in the range of [0.8, 0.9], Precision values of all methods are very high so average F-measure values are quite high. For the threshold value is equal to 0.7, our measure and Levenshtein's measure are only a little bit different in term of F-measure value: 0.8373 and 0.8377, respectively. In other cases, our measure is the best one.

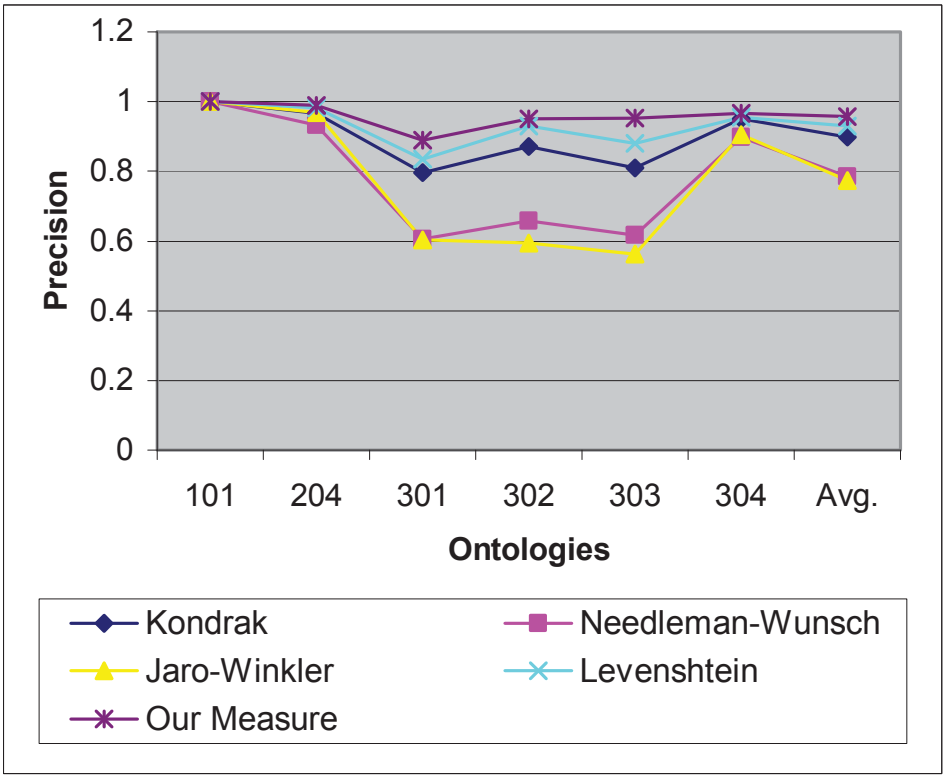

Figure 4.1: Average Precision of measures for six pairs of ontologies with different thresholds

Figures 4.1, 4.2, and 4.3 present average Precision, Recall and F-measure of measures for six pairs of ontologies with thresholds changed, respectively. As can be seen in Fig. 4.2, Recall values have only a little bit changed with all measures. Therefore, F-measure have been changed when Precision values have been changed. In figures 4.1 and 4.3, Precision and F-measure obtained by Levenshtein and our approaches are higher than those of the other measures in general, so these two methods are used to make a more detailed comparison.

It is clear that our measure depends on parameters α and β due to its derivation from information-theoretic approach. To determine the range of parame-

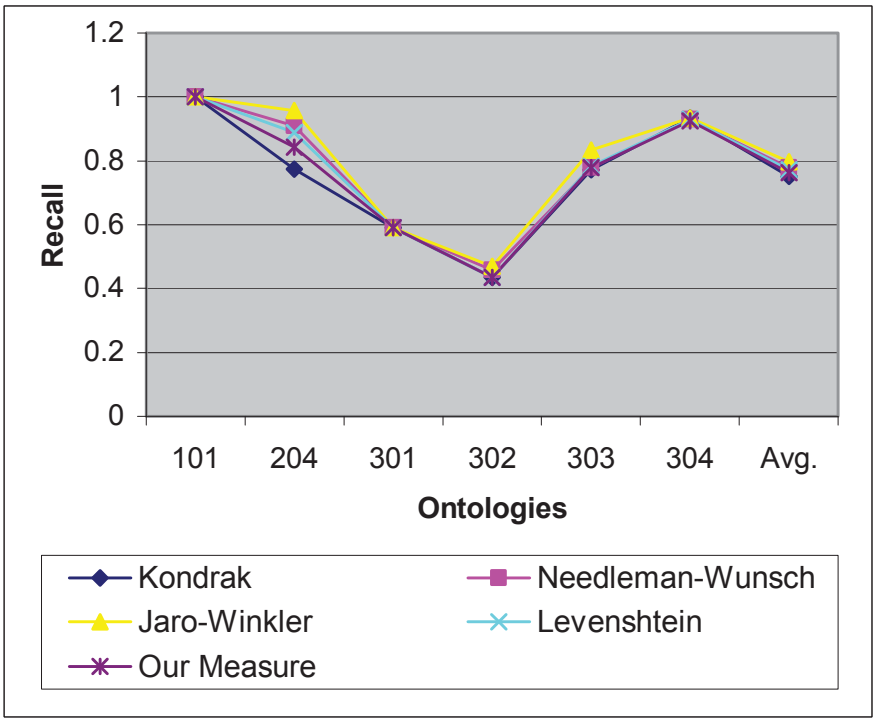

Figure 4.2: Recall of measures for six pairs of ontologies with different thresholds

ters in our measure which could obtain good results, parameter α changed for six different values from 0.2 to 0.7 with an increment of 0.1. The average Precision, Recall and F-measure of these two measures for six pairs of ontologies with thresholds and these parameters are presented in Table 4.3.

Table 4.3: Average Precision, Recall and F-measure values of two measures for six pairs of ontologies with an increment of parameters of 0.1 (Pre.=Precision, Rec.=Recall, F.=F-measure).

Average	Levenshtein	Our Measure					
		$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$
Precision	0.930	0.957	0.927	0.896	0.863	0.828	0.787
Recall	0.771	0.762	0.773	0.782	0.787	0.793	0.804
F-measure	0.832	0.836	0.833	0.825	0.815	0.803	0.790

The results show that increasing parameter α leads to our Precision value decreasing and our Recall value increasing. When $\alpha = 0.5$, our measure is similar to Dice's measure. However, our F-measure is lower than Levenshtein's. In the following experiment, parameter α takes values from the interval [0.2, 0.4] with an increment of 0.05. The results of average Precision, Recall and F-measure of our

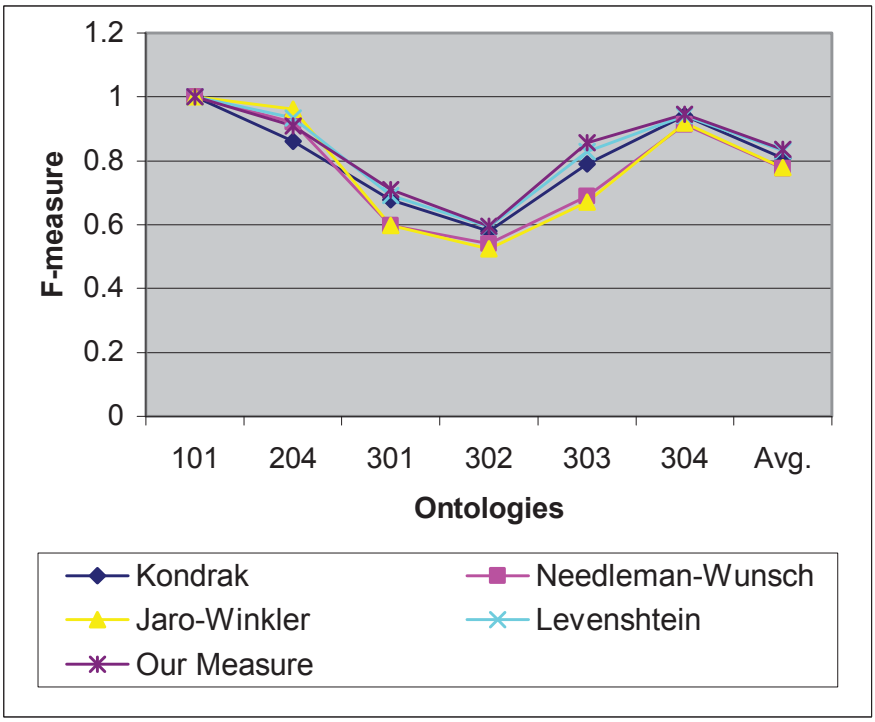

Figure 4.3: F-measure of measures for six pairs of ontologies with different thresholds

measure for six pairs of ontologies with these parameters are described in [86]. Figures 4.4, 4.5, and 4.6 represent average Precision, Recall and F-measure of two measures for six pairs of ontologies with thresholds and parameters changed, respectively. In Fig. 4.5, Recall values are almost the same. In this case, Precision values are the higher the better. As can be seen in Fig. 4.4, the higher parameter α is, the lower the Precision value is. To obtain good Precision values, parameter α should be chosen between 0.2 and 0.35. Consequently, β is in range from 0.4 to 0.325. Moreover, Precision values obtained by our method in this range are quite stable when compared to Levenshtein's measure.

Besides the above evaluation, our measure is also more rational in several cases. For example, given two strings c_1="glass" and c_2="grass". There is only one edit transforming c_1 into c_2: the substitution of "l" with "r". Therefore, the Levenshtein distance between two strings *glass* and *grass* is 1. Applying Eq. (3.10) and Eq. (4.10), the similarity between two strings *glass* and *grass* is 0.8 while the similarity degree of our measure yields 0.5. In fact, the two strings *glass* and *grass* describe different objects. While the Levenshtein measure returns the height similarity score value (0.8), the result 0.5 of our measure is quite reasonable. In another example, if $n \geq 2$ then two strings *Rep* and *Rap* have no n-grams in com-

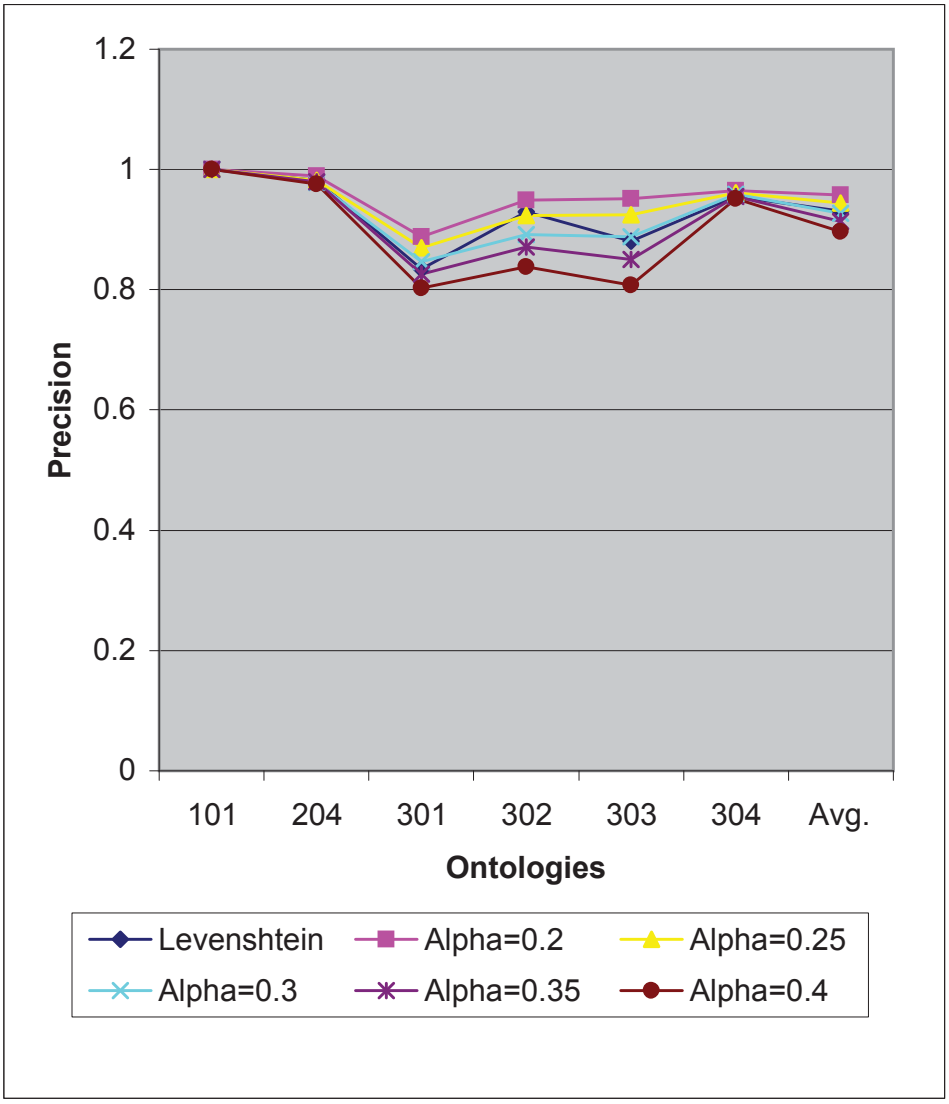

Figure 4.4: Precision of two measures for six pairs of ontologies with different thresholds and parameters

mon. In this case, applying Dice's measure to these strings brings the dissimilarity. Additionally, the family of Dice's methods has a characteristic which relies on the set of samples but not on their positions. Since the sets of bigrams of two strings *Label* and *Belab* including *{la, ab, be, el}* are the same, the similarity value of these strings equal to 1, which seems inappropriate. In short, our approach overcomes the limits of these cases.

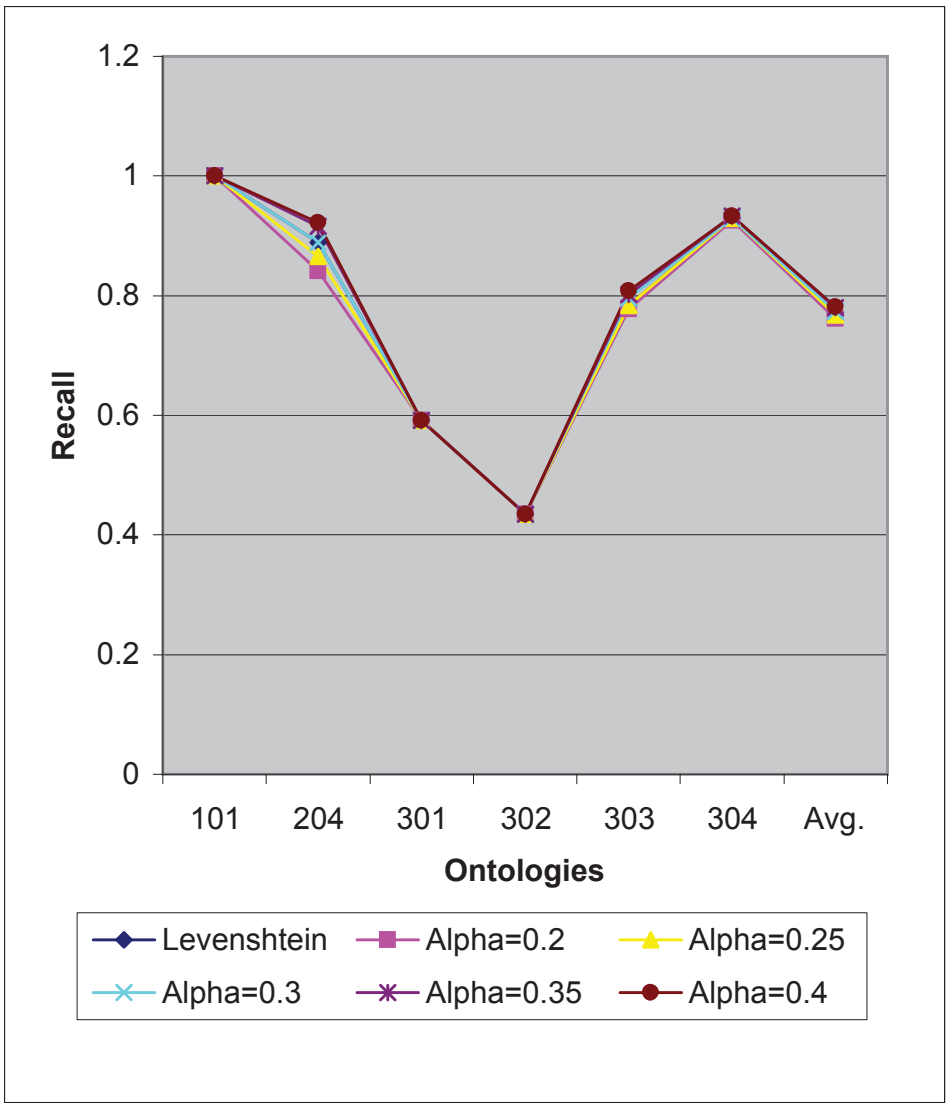

Figure 4.5: Recall of two measures for six pairs of ontologies with different thresholds and parameters

4.4. CONCLUSIONS AND FUTURE WORK

In this chapter, a new lexical-based approach was proposed, which considered the similarity of sequences by combining feature-based and element-based measures. This approach is motivated by Tversky's and Levenshtein's measures; however, it is completely different from original lexical methods previously presented. The main idea of our approach is that the similarity value of two given concepts depends not only on the contents but also on the editing operations of these concepts in strings. For Levenshtein's measure, it focus on the number of edit-

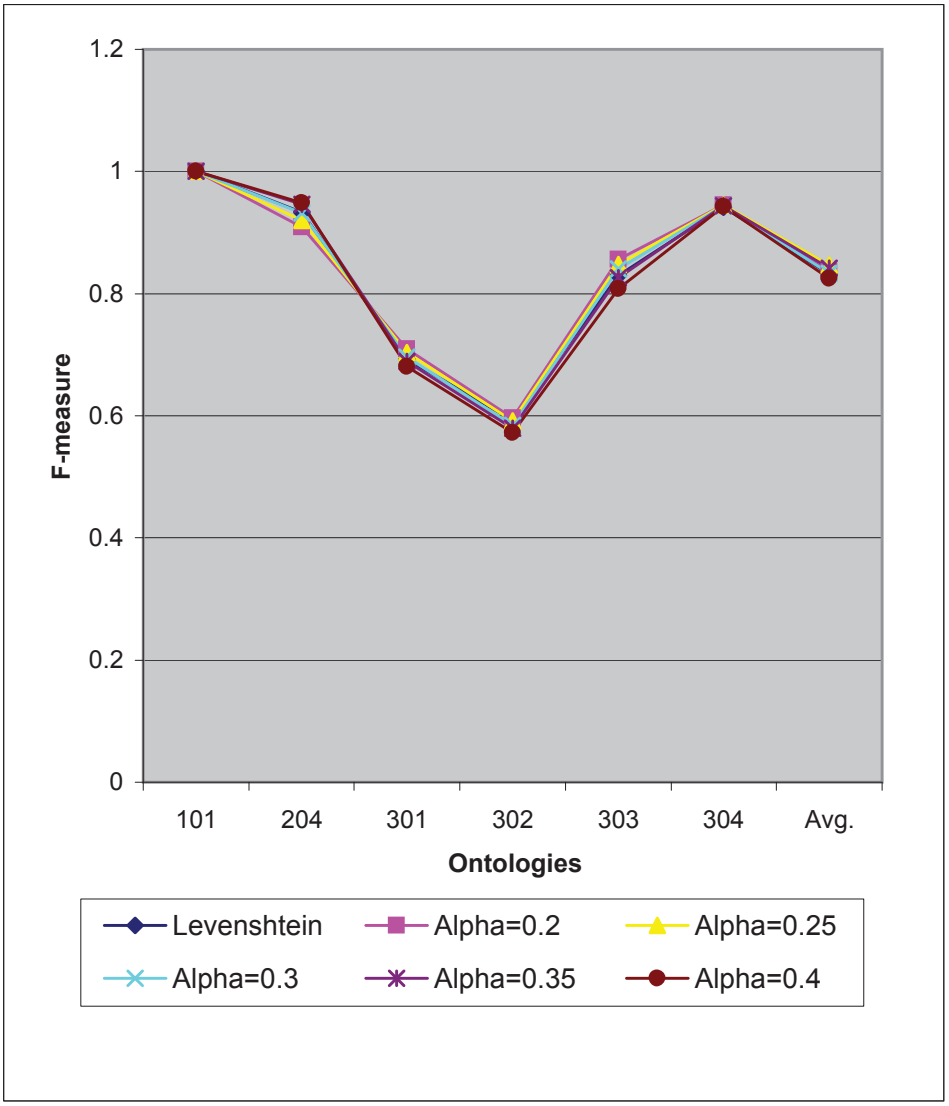

Figure 4.6: F-measure of two measures for six pairs of ontologies with different thresholds and parameters

ing operations in order to change one string into another string. For Tversky's model, the more common features and the less different features are, the higher the similarity values between objects are obtained. For this reason, the combination of the two above models reduces the limitations of other methods. The experimental validation of the proposed metric has been conducted through six pairs of ontologies in the benchmark dataset of the 2008 OAEI, and compared to four common similarity metrics including Jaro-Winkler, Needleman-Wunsch, Kondrak and Levenshtein metrics. The results show that our sequence similarity metric provides good values compared to other existing metrics. Moreover, our

metric can be considered as a general and flexible lexical approach. In particular, adjusting the parameters α and β produces the popular measures making convenient experiments. It can also be implemented in many domains in which strings are short such as labels of concepts in ontologies, proteins and so on.

In this work, strings are considered as a set of characters. However, they can be extended to the set of tokens in which the similarity between chunks in plagiarism detection is calculated. Besides, our string-based similarity metric might also be combined with relations between entities in ontologies using WordNet dictionary to improve the semantic similarity of pairs of these entities.

4.5. SUMMARY

This chapter illustrated the combination of information-theoretic and edit distance measures. Additionally, properties of the proposed measure which need to satisfy the characteristics of similarity functions are given. Some experimental results and discussions were presented in this chapter.

In the next chapter of this thesis, a new approach, which exploit the semantic information to find correspondences is introduced.

5

Semantic Similarity Measure between Nouns based on the Structure of WordNet

Several approaches for computing semantic similarity and relatedness measures between terms have been developed. This chapter proposes a new semantic similarity measure between two nodes concentrating on nouns as well as their hypernym/hyponym relationships based on the structure of WordNet. In particular, the similarity of two given nouns depends not only on their positions in the hierarchy but also on their relevancy connections. Moreover, the characteristics of this method are that it is based on edge-counting and does not need a large corpus so it is computationally efficient.

The remainder of this chapter is organized as follows. Section 5.1 introduces semantic similarity methods and the major idea of our approach. In section 5.2, our semantic similarity measure taking into account nouns of WordNet hierarchy and hypernym/hyponym relationships is explained. In section 5.3, we report the results formed by applying our approach on Miller and Charles benchmark test and give an evaluation of our measure and compare with other approaches and human similarity judgements. Later on, conclusions and future work are shown in section 5.4. Finally, a summary of this chapter is reviewed in section 5.5.

5.1. INTRODUCTION

Semantic similarity or semantic relatedness measures between concepts are widely applied in information retrieval and natural language processing such as spelling correction, word sense disambiguation and question answering. A number of semantic relatedness approaches as well as semantic similarity approaches calculating the similarity between terms have been proposed so far. These approaches can be classified based on part of speech (e.g. nouns, verbs, adjectives and adverbs), relations (e.g. hypernym/hyponym, meronym/holonym and antonym), different methods (e.g. information content, structure, feature and hybrid methods), or large information sets (e.g. WordNet and Wikipedia). In this chapter, similarity measures based on characteristics of these methods are presented. Generally speaking, these semantic similarity methods organized in a hierarchy can be grouped into four main categories. These categories include path length-based methods (so-called edge counting methods), information content-based methods (also called node-based methods), feature-based methods, and hybrid methods. The main objective of these approaches is to determine the degree of semantic similarity between two words for matching human judgements as closely as possible. A brief characteristic of these methods is presented as follows:

- Path length-based measures use the structure of semantic networks. In these methods, the lengths of links are considered the same while the densities of nodes are ignored. These approach are determined by path length from one concept to another concept and depths of these concepts in the hierarchy tree [67, 70, 103, 142]. Consequently, the shorter the distances between concepts are calculated, the more similar they are. These methods have low computational cost [113] because they implement counting edges to obtain the similarities of pairs of nodes. However, these methods only concentrate on the minimum paths and do not take care of the relationships between nodes leading to coarse results.

- Information content-based measures employ the notion of information content. These approaches are computed by counting the occurrences of words in a large corpus [71, 105]. Therefore, the larger the size of the corpus is, the more precise the probabilities of occurrence of concepts are obtained but the more computation time is needed. The characteristics of these methods could be summarized as follows. Firstly, the probability of occurrence of a concept depends on probabilities of appearances of

its descendants in the hierarchy. Secondly, subsumers are considered as more abstract than their subsumed concepts, so they contain less information content than their children. For that reason, the leaves bring the most amount of information. Finally, the information contents of leaves are the same. Therefore, some similarity measures, for example Lin [71] and Jiang&Conrath [59], return the same similarity between all pairs of leaves in case these nodes share a nearest common ancestor. In this approach, the similarities between concepts are the amount of shared information between these concepts. In other words, the more the common information of concepts is, the higher the similarity degree of these concepts is. The disadvantages of this approach is that in case the taxonomy is changed, the similarity values become different. Consequently, some measures are proposed to calculate the information content based on intrinsic information content in the specific ontologies [49, 115, 128].

- Feature-based measures focus on properties or sets of glossary in large data resources such as Wikipedia or WordNet. The similarities of pairs of concepts are considered a function of attributes, so these approaches do not take care of the lengths of links. However, they ignore information on the structure hierarchy [110, 114, 133, 136]. Feature-based measures usually employ Tversky's model [133] based on the common and distinct characteristics of concepts.

- Hybrid methods combine multiple information sources and other measures [59, 106].

In fact, the objectives of applications relating to searched and queried information are that the returned results should be relevant to the user's queries and computation time as short as possible. Therefore, we need to apply a technique taking the advantage with respect to the computation cost. Additionally, a structure in the taxonomy usually brings a lot of semantics. From the state of the review described above, we choose path length-based technique to form our measure. To improve accuracy of similarity values, however, our measure is based on features of semantic networks. Our approach takes into account both their positions and relevant relationships between considered concepts which is different from other edge counting methods previously presented in section 3.2. Nowadays, WordNet is considered as a background knowledge source used in natural language processing and computational linguistics so we use WordNet to take semantics of concepts.

In this study, a semantic similarity metric calculating the similarity between two concepts in the taxonomy of WordNet dictionary using path length-based technique is proposed. Particularly, our metric focuses on the similarity values between nouns and hypernym/hyponym relations based on the structure of WordNet hierarchy. Fig. 5.1 shows a fragment of WordNet hierarchy where a virtual root node at top level, hypernym of all concepts, is added. Since hy-

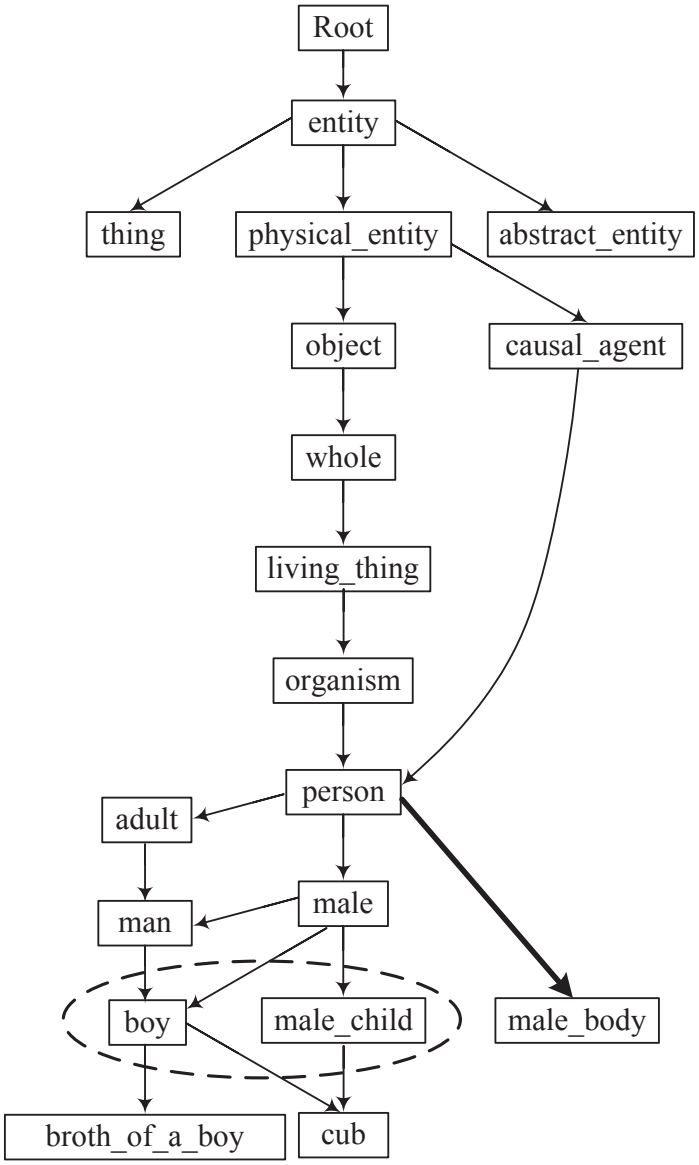

Figure 5.1: A fragment of the WordNet nouns taxonomy. Single lines indicate *is-a* links, thick lines represent *part-of* links, the dash ellipse depicts a synset

pernym/hyponym links account for about 80% of all relation types in WordNet,

we here concentrate on calculating the semantic similarity between nouns along these relations. The main idea of the approach is that the similarity value of two given concepts depends not only on the positions of these concepts in the hierarchy but also on their semantic relationships. Our method is partly motivated by Wu&Palmer's measure [142] and adds some more relationships relating to the compared concepts. According to [71], Wu&Palmer's measure has a simple implementation and good performances compared to the other measures. For that reason, this method is chosen in this study. Moreover, our approach is different from other ones because it does not depend on a large text corpus or glosses in the structure which contains a lot of information.

5.2. SIMILARITY MEASURE BASED ON EDGE-COUNTING (sim_{NC})

In the current section, the conceptual similarity between nodes is presented. Moreover, two definitions including the direct and indirect connections between two nodes are described in subsection 5.2.1. The similarity of a pair of concepts in the structured databases is usually considered based on four intuitions expressed in subsection 5.2.2. From these discussions, a new edge-counting similarity measure calculating semantic similarity between two given concepts is proposed. A detailed description is shown in subsection 5.2.3.

5.2.1. BASIC DEFINITIONS

Let C_i, C_j, C_k be arbitrary distinct nodes in WordNet hierarchy.

Definition 1. The *direct connection* from C_i to C_j is a directed path from C_i to C_j.

Note that with a complex semantic network as WordNet, there might be more than one direct connection between two nodes C_i and C_j in which the lengths of connections can be different.

Definition 2. Let C_k be a common descendant of C_i and C_j. The *indirect connection* between C_i and C_j through C_k is a compound path of two direct connections that includes the direct connection from C_i to C_k together with the direct connection from C_j to C_k.

Without loss of generality, relevant relationships between two nodes C_4 and C_5 are presented as in Fig. 5.2 where:

- $C_4 \rightarrow C_6 \rightarrow C_9$ is a direct connection from node C_4 to node C_9

- $C_4 \rightarrow C_7 \rightarrow C_{10} \leftarrow C_{12} \leftarrow C_{11} \leftarrow C_5$ is an indirect connection between C_4 and C_5 through the node C_{10}.

5.2.2. INTUITIONS

Before calculating the semantic similarity of a pair of nodes, four intuitions are introduced.

Intuition 1. *In the hierarchy, a node has its ancestors and descendants. Besides, there are direct and indirect connections between two considered concepts as well as connections from ancestors of one node to ancestors and descendants of the other node and from descendants of one node to ancestors and descendants of the other node. As can be seen in Fig. 5.2, there are one direct path and several indirect paths connecting two nodes C_4 and C_5. Furthermore, C_4 is the descendant of C_1 and C_5 is the ancestor of C_{12}, and there exists a path from C_1 to C_{12}. Therefore, the semantic similarity of nodes depends not only on themselves, their direct and indirect relations, but also on connections relating to these nodes.*

Intuition 2. *Consider a hierarchical taxonomy including concepts and hyponym/hypernym links. In case the distances of pairs of concepts are equal, two concepts which belong to an upper level should be less similar than those of a lower level [106, 142] since concepts on higher levels are less detailed than those on lower levels.*

Intuition 3. *The similarity of a pair of nodes depends on the number of links from one node to the other one. It means the longer the path between the nodes is, the more their semantic similarity decrease [59].*

Intuition 4. *Normally, the calculation of the similarity value should be considered with the weight of each link. Some researchers agree that the weight of a link depends on factors such as the depth of concepts, the density and the link types [106, 131]. However, in this thesis we assume that the weights of "is-a" links are the same.*

5.2.3. PROPOSED MEASURE

In this subsection, our semantic similarity measure relying on the previous intuitions is presented in detail. Here, our measure adopts nouns and their relations in WordNet taxonomy.

The connections relating to the two noun nodes C_i and C_j include:

- Direct connections between C_i and C_j;
- Connections between C_i and ancestors of C_j as well ancestors of C_i and C_j;
- Connections between C_i and descendants of C_j as well descendants of C_i and C_j;
- Connections between ancestors of C_i and ancestors of C_j as well descendants of C_i and descendants of C_j;
- Connections between ancestors of C_i and descendants of C_j as well descendants of C_i and ancestors of C_j;

To obtain the similarity of two nodes, Wu&Palmer's measure is applied as starting point. Let $sim_{WP}(C_i, C_j)$ be the similarity value of C_i and C_j based on Wu&Palmer's method, the length of a connection is the number of links, $length(C_i \rightarrow C_j)$ is length of direct connection from C_i to C_j, $length(C_i \rightarrow C_k \leftarrow C_j)$ is length of the indirect connection of $C_i \rightarrow C_k \leftarrow C_j$, w is the weight of "is-a" link.

Now, an example for calculating the similarity between the concepts C_4 and C_5 in Fig. 5.2 is given. Note that the indirect connections between nodes through their descendants in which the lengths of the direct connections from these nodes to their own descendants are equal and greater than 2 (for example $C_4 \rightarrow C_7 \rightarrow C_{10} \leftarrow C_{12} \leftarrow C_{11} \leftarrow C_5$), the connections of their ancestors (for example $C_2 \rightarrow C_3$), the connections of their descendants (for example $C_{12} \rightarrow C_{10}$), the connections between C_4 and ancestors of C_5 as well ancestors of C_4 and C_5 (for example $C_3 \rightarrow C_4$, $C_3 \rightarrow C_5$, respectively), the connections between the ancestors of one node and descendants of the other node (for example $C_1 \rightarrow C_{12}$) bring about only slight effects on the similarity values as well as lead to a bad performance. Therefore, at this time, those connections are ignored. The similarity of a pair of nodes here is computed based only on themselves, their direct connections and their indirect connections at their children. The similarity between two concepts C_4 and C_5 depends on the following connections:

- Direct connection between C_4 and C_5, that is: $C_4 \rightarrow C_5$;
- Connections between C_4 and descendants of C_5 through the children of C_4 as well as descendants of C_4 and C_5 through the children of C_5, that are: $C_4 \rightarrow C_8 \leftarrow C_5$, $C_4 \rightarrow C_6 \rightarrow C_9 \leftarrow C_5$.

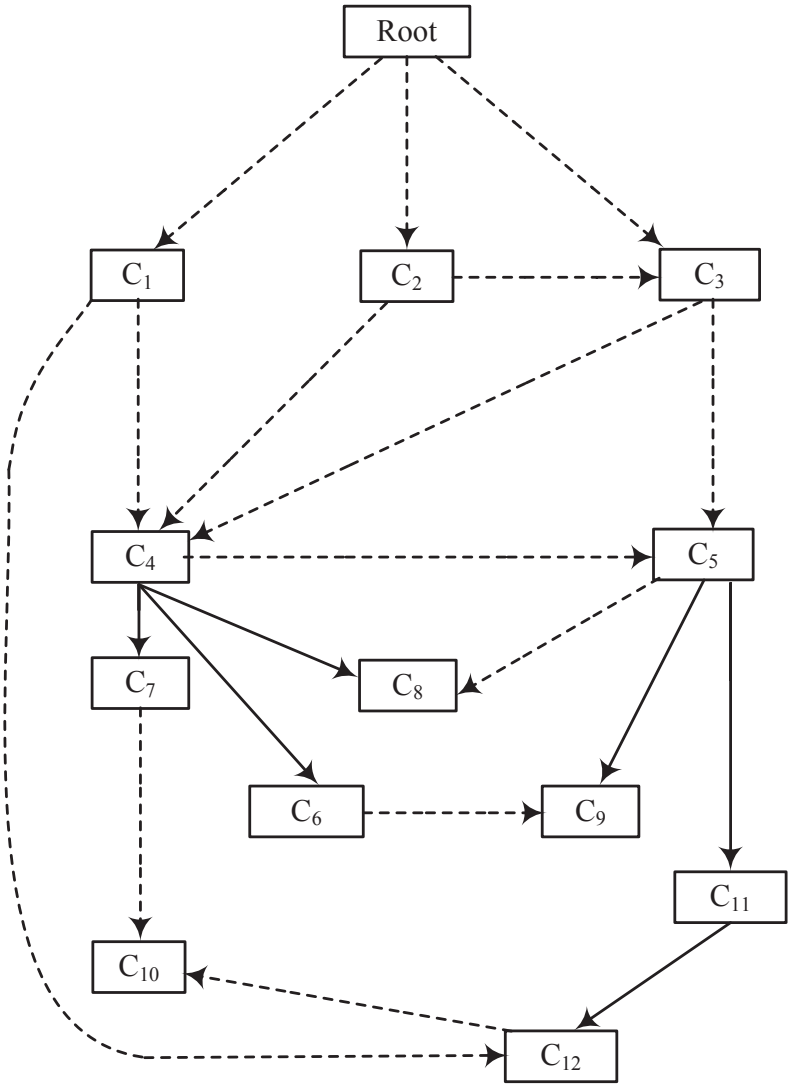

Figure 5.2: Relevancy connections *is-a* between C_4 and C_5. Single line indicates *is-a* link, dashed lines represent one or more *is-a* links

The similarity between C_4 and C_5 with respect to their direct connection is determined as follows:

$$\delta_{connect\_1} = sim_{WP}(C_4, C_5) * w^{length(C_4 \rightarrow C_5)} \tag{5.1}$$

The similarity between C_4 and C_5 with respect to their indirect connection through the children of C_4 is determined as follows:

$$\delta_{connect\_2} = \sqrt{sim_{WP}(C_4, C_8) * sim_{WP}(C_8, C_5)} * w^{length(C_4 \rightarrow C_8 \leftarrow C_5)} \tag{5.2}$$

The similarity between C_4 and C_5 with respect to their indirect connection

through the children of C_5 is determined as follows:

$$\delta_{connect\_3} = \sqrt{sim_{WP}(C_4, C_9) * sim_{WP}(C_9, C_5)} * w^{length(C_4 \rightarrow C_9 \leftarrow C_5)} \quad (5.3)$$

After a set of components relating to the two concepts C_4 and C_5 is determined, the similarity between these nodes is calculated. This similarity measure results from the combination of component values and is given by:

$$sim(C_4, C_5) = \alpha * sim_{WP}(C_4, C_5) + (1-\alpha) * \frac{\delta_{connect\_1} + \delta_{connect\_2} + \delta_{connect\_3}}{\sqrt{3 * (\delta^2_{connect\_1} + \delta^2_{connect\_2} + \delta^2_{connect\_3})}} \quad (5.4)$$

Generally, $\delta_{connect\_t}$ is a component similarity of two concepts including the direct or indirect connections where *connect_t* is a positive integer less than or equal to number of the direct connections and indirect connections between two nodes C_i and C_j through their children. $\delta_{connect\_t}$ should depend on the length of connection and the weight of each link which is $w^{length(C_i \rightarrow C_j)}$ for a direct connection between C_i and C_j or $w^{length(C_i \rightarrow C_k \leftarrow C_j)}$ for an indirect connection between C_i and C_j through C_k. Because our approach is based on Wu&Palmer's measure, $\delta_{connect\_t}$ should depend on $sim_{WP}(C_i, C_j)$ in case there exists a direct connection from C_i to C_j. Besides, if there exists an indirect connection from C_i to C_j through C_k, $\delta_{connect\_t}$ should depend on $\sqrt{sim_{WP}(C_i, C_k) * sim_{WP}(C_k, C_j)}$. Therefore, $\delta_{connect\_t}$ value can be presented as

$$\delta_{connect\_t} = sim_{WP}(C_i, C_j) * w^{length(C_i \rightarrow C_j)} \quad (5.5)$$

if there exists a direct connection from C_i to C_j

and

$$\delta_{connect\_t} = \sqrt{sim_{WP}(C_i, C_k) * sim_{WP}(C_k, C_j)} * w^{length(C_i \rightarrow C_k \leftarrow C_j)} \quad (5.6)$$

if there exists an indirect connection from C_i to C_j through C_k.

At this time, the overall similarity of any pair of distinct concepts C_i and C_j is defined by the following formula

$$sim_{NC}(C_i, C_j) = \alpha * sim_{WP}(C_i, C_j) + (1-\alpha) * \frac{\sum_{t=1}^{n} \delta_{connect\_t}}{\sqrt{n * \sum_{t=1}^{n} \delta^2_{connect\_t}}} \quad (5.7)$$

where n is the number of the direct connections and indirect connections between two nodes C_i and C_j through their children, α is an adjusted parameter in range from 0 to 1.

According to Eq. 5.7, the similarity value of two arbitrary nodes takes values from the interval [0, 1]. The similarity of two distinct nodes is equal to 1 if and only if $sim_{WP}(C_i, C_j) = 1$, i.e., the two nodes belong to a synset, and the connections between two nodes have to exist and the similarities of these component connections are the same. For example, considering *gem* node and *jewel* node, although they belong to the same synset, the similarities of their component connections, e.g. $jewel \rightarrow diamond \leftarrow gem$ and $jewel \rightarrow sapphire \leftarrow transperent\_gem \leftarrow gem$, are different leading to the similarity value smaller than 1. This is completely suitable for human measurement.

To illustrate the main idea of our method, an example for computing the similarity between two nodes *Food* and *Fruit* in WordNet taxonomy is presented. Fig. 5.3 depicts a fragment of relationships between *Food* and *Fruit* concepts.

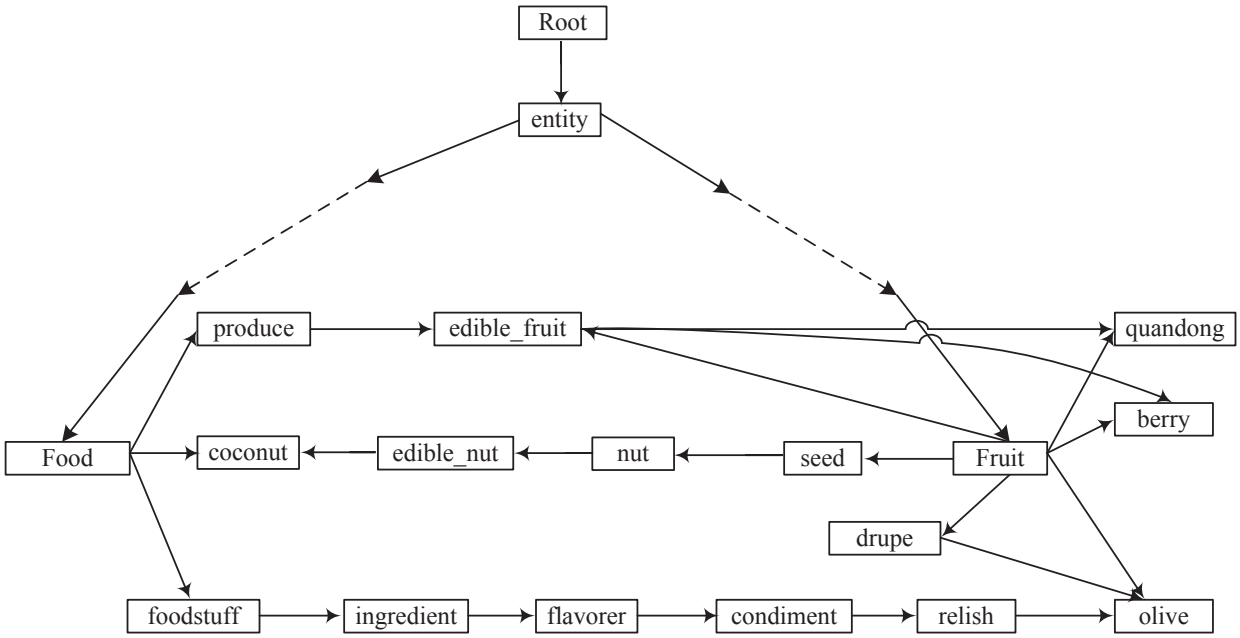

Figure 5.3: A fragment of relationships between *Food* and *Fruit* concepts in WordNet taxonomy

Step 1. Check the direct connections between two nodes *Food* and *Fruit*. If they exist, save them.

In this example, there are no direct connections between *Food* and *Fruit* concepts in WordNet hierarchy.

Step 2. Find all children of *Food* node. For each child of *Food*, find the di-

rect connections from *Fruit* node to the children of *Food*, which are the indirect connections through the children of *Food* node. If they exist, save them.

In this step, we get one connection as follows.

$connect\_1 : Food \rightarrow coconut \leftarrow edible\_nut \leftarrow nut \leftarrow seed \leftarrow Fruit$;

Step 3. Find all children of *Fruit* node. With each child of *Fruit*, find the direct connections from *Food* node to the children of *Fruit*, which are the indirect connections through the children of *Fruit* node. If they exist, save them.

There are four connections found in this step, these are:

$connect\_2 : Fruit \rightarrow edible\_fruit \leftarrow produce \leftarrow Food$;

$connect\_3 : Fruit \rightarrow quandong \leftarrow edible\_fruit \leftarrow produce \leftarrow Food$;

$connect\_4 : Fruit \rightarrow olive \leftarrow relish \leftarrow condiment \leftarrow flavorer \leftarrow ingredient \leftarrow foodstuff \leftarrow Food$;

$connect\_5 : Fruit \rightarrow berry \leftarrow edible\_fruit \leftarrow produce \leftarrow Food$.

In both step 2 and step 3, the connections are obtained, which are necessary for the similarity between *Food* and *Fruit* nodes.

Step 4. Compute the component similarities between *Food* and *Fruit* nodes corresponding to the connections:

$$\delta_{connect\_1} = \sqrt{sim_{WP}(Food, coconut)} * \sqrt{sim_{WP}(coconut, Fruit)} * w^{length(connect\_1)} \tag{5.8}$$

$$\delta_{connect\_2} = \sqrt{sim_{WP}(Fruit, edible\_fruit)} * \sqrt{sim_{WP}(edible\_fruit, Food)} * w^{length(connect\_2)} \tag{5.9}$$

$$\delta_{connect\_3} = \sqrt{sim_{WP}(Fruit, quandong)} * \sqrt{sim_{WP}(quandong, Food)} * w^{length(connect\_3)} \tag{5.10}$$

$$\delta_{connect\_4} = \sqrt{sim_{WP}(Fruit, olive)} * \sqrt{sim_{WP}(olive, Food)} * w^{length(connect\_4)} \tag{5.11}$$

$$\delta_{connect\_5} = \sqrt{sim_{WP}(Fruit, berry)} * \sqrt{sim_{WP}(berry, Food)} * w^{length(connect\_5)} \tag{5.12}$$

Step 5. Calculate the similarity between two nodes *Food* and *Fruit* applying our measure

$$sim_{NC}(Food, Fruit) = \alpha * sim_{WP}(Food, Fruit) + (1-\alpha) * \frac{\sum_{t=1}^{5} \delta_{connect\_t}}{\sqrt{5 * \sum_{t=1}^{5} \delta^2_{connect\_t}}} \tag{5.13}$$

In the following subsection, the properties of our semantic similarity measure Sim_{NC} are discussed.

5.2.4. PROPERTIES OF OUR SEMANTIC SIMILARITY MEASURE

Our measure presented in Eq. (5.7) satisfies properties of a similarity measure as follows.

- Positiveness: $\forall c_i, c_j : Sim_{NC}(c_i, c_j) \geq 0$

  Proof. We have $Sim_{WP}(c_i, c_j), w \geq 0$, the lengths between c_i and $c_j \geq 0$, and $1 \geq \alpha \geq 0$. Therefore, $Sim_{NC}(c_i, c_j) \geq 0$. □

- Maximality: $\forall c_i, c_j, c_t : Sim_{NC}(c_t, c_t) \geq Sim_{NC}(c_i, c_j)$

  Proof. Because $Sim_{WP}(c_i, c_j)$ and $w \leq 1$, the similarity between two distinct concepts c_i and $c_j \leq 1$.
  Moreover, we have $Sim_{NC}(c_t, c_t) = 1$.
  Therefore, $Sim_{NC}(c_t, c_t) \geq Sim_{NC}(c_i, c_j)$. □

- In case all connections are used, the proposed measure is symmetric. However, to reduce the computational time, some connections are ignored. Therefore, at that time our measure is not symmetric.

Experimental results as well as evaluations are presented in the following section to get a better overview for our method.

5.3. EXPERIMENTAL RESULTS

WordNet version 2.1 [4] is chosen to implement experiments. A free online package developed by Ted Pederson et al. [98] is used to obtain the similarity values between entities when different semantic similarity methods are applied. All semantic similarity measures that we chose here based on WordNet dictionary are Rada [103], Wu&Palmer [142], Lin [71], Context Vector [136], Gloss vectors [97] and Pairwise [96]. Many researchers also showed their results using the same pairs of concepts in datasets. In fact, the datasets are usually used for evaluating semantic similarity measures of words extracted from the practical data of Rubenstein and Goodenough [111] as well as Miller and Charles [76]. These datasets were estimated by humans obtaining the similar meaning values of one word to an other one.

The benchmark test by Miller and Charles is used for our method and then our results are compared with the other measures based on the correlation coefficients of the similarity values against human judgements.

Since our measure depends on parameters α and w, these parameters are adjusted to calculate semantic similarity values of all of the concept pairs on Miller and Charles dataset. After that, the correlation coefficient is calculated corresponding to each pair of the parameter value. By changing the values of the parameters, correlation coefficients are obtained. The parameters which produce the best value of correlation coefficient will be chosen. Our measure with different values of α and w is tested and then $\alpha = 0.7$ and $w = 0.9$ are chosen. Table 5.1 shows semantic similarity values applying our measure based on optimal parameters on Miller and Charles dataset.

Figures 5.4 and 5.5 show two charts with the similarities obtained by human ratings and different methods, respectively in which the human ratings were evaluated from 0 to 4 and the similarity values represented in Fig. 5.5 are in the range [0,1].

Because the units in which the similarity values are given are not uniform, a method used in comparison is applied similarly to [5]. In particular, scaling these similarity values obtained by the considered approaches in the range [0,1] to [0,4]. This leads to an easier comparison among measures and human ratings. Fig. 5.6 illustrates the similarity values scaled to [0,4].

The method proposed by Wu&Palmer is used to obtain initial similarity weights which we later enhance based on our measure. Therefore, we first compare our

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

Table 5.1: Semantic similarity values applying our measure on Miller and Charles dataset.

No.	Word pairs	Sim_{NC}	No.	Word pairs	Sim_{NC}
1	car - automobile	0.9998	16	lad-brother	0.5714
2	gem - jewel	0.9983	17	journey-car	0.1524
3	journey - voyage	0.9567	18	monk-oracle	0.4706
4	boy - lad	0.7466	19	cemetery-woodland	0.4
5	coast - shore	0.9385	20	food-rooster	0.2286
6	asylum - madhouse	0.7652	21	coast-hill	0.5714
7	magician - wizard	0.9999	22	forest-graveyard	0.4
8	midday - noon	0.8	23	shore-woodland	0.5334
9	furnace - stove	0.4571	24	monk-slave	0.5714
10	food - fruit	0.5645	25	coast-forest	0.4923
11	bird-cock	0.9496	26	lad-wizard	0.5714
12	bird-crane	0.8972	27	cord-smile	0.3
13	tool-implement	0.9488	28	glass-magician	0.4266
14	brother-monk	0.9652	29	noon-string	0.2823
15	crane-implement	0.6222	30	rooster-voyage	0.1185

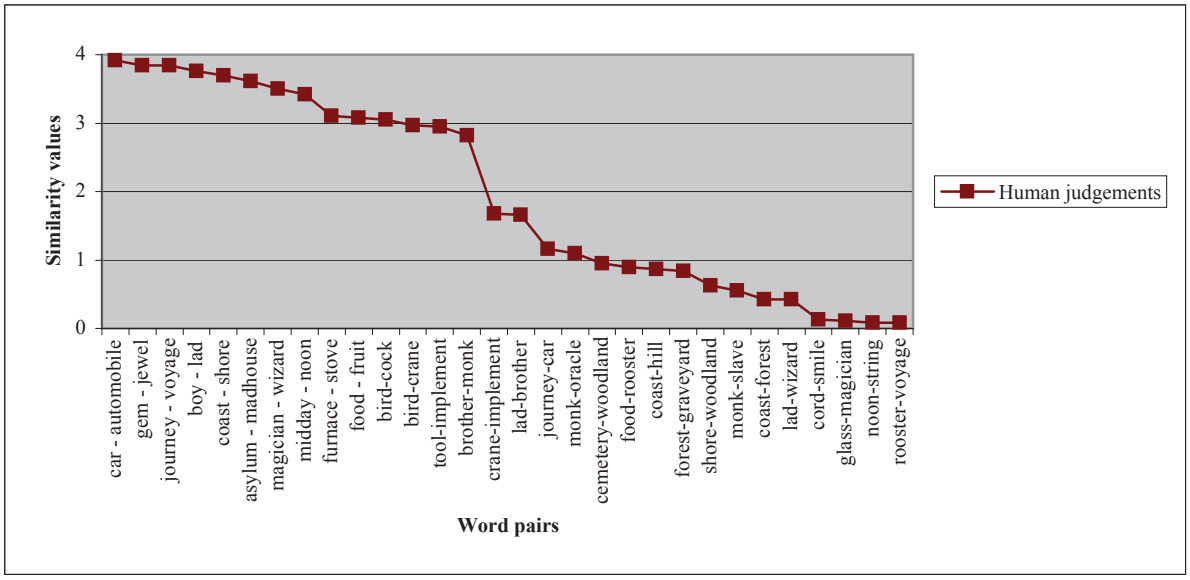

Figure 5.4: Human judgements

results with the results applying Wu&Palmer method. It is possible to see in Fig. 5.6 that the trend of the curve in the chart based on Miller and Charles dataset

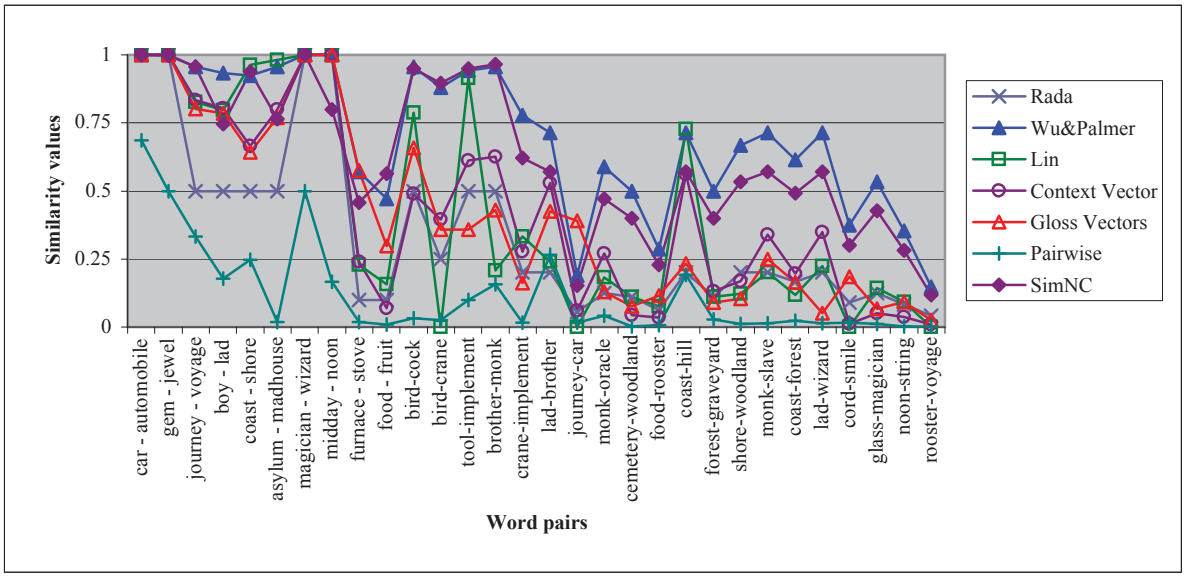

Figure 5.5: Different measures

5

is descending. The curve indicating our measure is nearer than the approach of Wu&Palmer compared to human judgements. That means our method is more accurate than Wu&Palmer's.

In order to compare our results with the other measures, the following discussion is carried out. First of all, as shown in Fig. 5.6, the Gloss vectors curve seems to be nearest in comparison with human ratings curve. Secondly, the deviations of the first 20 pairs applying our method and Gloss vectors method are highly similar. However, for the last 10 pairs, deviation of our curve is far from human curve. The reason is that the pairs of nodes at low level are classified in distinct subgraphs. As a result, there are a few relevant relations between the considered concepts. On the other hand, our results are quite similar to those of Wu&Palmer's approach in these cases.

For the other methods, the similarity values of concepts at the low level are close to human ratings. However, the curves presenting the similarities of concepts at the high and immediate levels have a large variation.

Next applying the correlation coefficients formula brings correlation values between the human judgement and different measures. Our result is shown in the first row in Table 5.2.

From the results in Table 5.2, the gloss based methods obtain relative high correlation coefficients. In fact, they are based on the glosses in WordNet which are often too short and not enough to provide vocabularies. Thus, some ap-

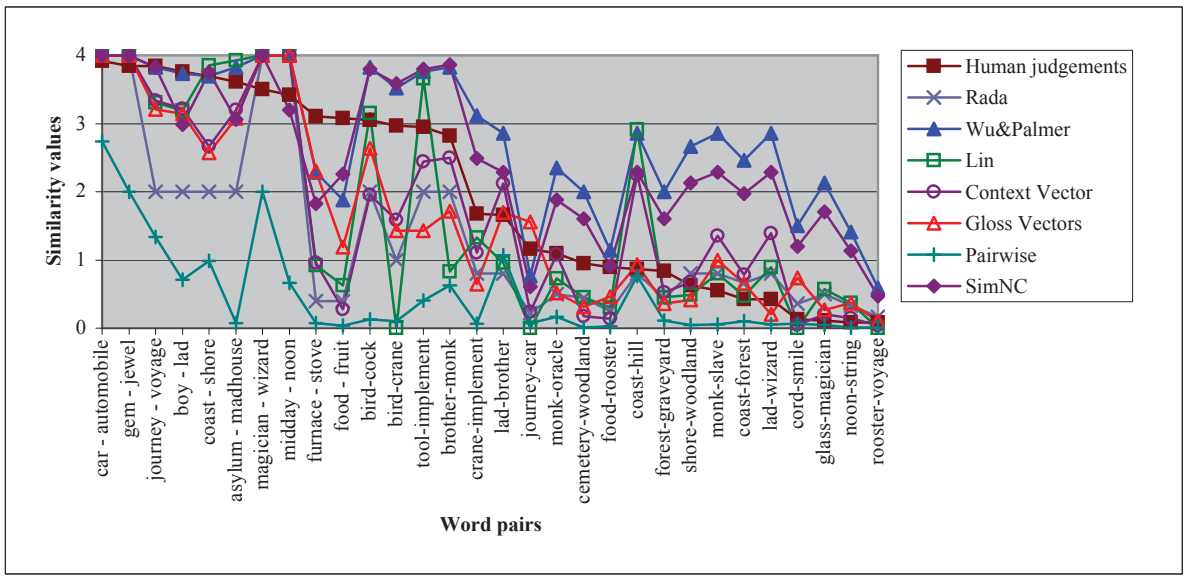

Figure 5.6: Human judgements and different measures after being scaled

5

Table 5.2: Correlation coefficients between human judgements and different measures.

Semantic Similarity Measures	Correlation Coefficients
Sim_{NC}	0.825
Wu&Palmer	0.768
Rada	0.755
Lin	0.770
Pairwise	0.605
Context Vector	0.807
Gloss Vectors	0.885

proaches extended either related synsets or glosses of related concepts to get more accurate similarity values. Methods based on the information content depend on size of corpus for estimating the frequency of words. The edge-based methods consider the position of concepts in hierarchy but they omit the relationships between nodes. Therefore, the similarities are not high enough. It is clear that the correlation coefficient obtained by our measure is better than both path length-based and information content methods.

According to the analysis above, it can be seen that our method is reasonable. Our method is based on edge-counting technique. Furthermore, the relations of concepts corresponding to related nodes are considered. Thus, the correlation

of our measure for human similarity ratings is better compared to other edge-counting approaches. Besides, it is one of the best correlation coefficients compared to other methods.

5.4. CONCLUSIONS AND FUTURE WORK

In this chapter, an approach for measuring the semantic similarity between two nouns based on the structure of WordNet dictionary without the dependence on any large dataset, additional information resources and preprocessing data was proposed. Moreover, our measure was developed from the edge-counting approach which has relatively low computational cost. From results compared with other approaches, the correlation coefficient to the human judgements obtained by our approach is relatively high. It is better than the correlation coefficients of other edge-based methods. Moreover, our approach is also one of the best systems. Although this model intends to apply for ontology matching, it could be applied to multimedia objects, for example, images and videos by using captions and titles of these objects. Besides, it can be applied in various domains and used in combinations of approaches in order to come up with better correlation coefficients.

Because of its simplicity, we will calculate the semantic relatedness measure based on the combination of edge counting-based, information content-based and feature-based techniques as well as use other attributes of the taxonomy and other types of relationship such as meronym/holonym relation without compromising the generality of the method. Furthermore, different weights depending on kinds of relationships and positions of terms in the hierarchy can be assigned. Besides, the similarity of compound nouns appearing often in ontologies will be also invested.

5.5. SUMMARY

In this chapter a classification of semantic similarity measures, the characteristics, and the disadvantages of these approaches were presented and then an overview of the main idea of our measure was also described. Some basic definitions, intuitions, and the proposed method which reduces the restrictions of other methods, have been explained in detail. Furthermore, an illustrative example has been introduced to explain more clearly our measure. Our measure and the other ones were evaluated on dataset of Miller-Charles and then the correla-

tion coefficients to the human judgements were computed. We also compared the result applying our measure with those of other methods. The experimental results show that our method outperforms edge-counting methods. Finally, this chapter presented some conclusions and future directions. A novel structural similarity measure integrated by the lexical and structure measures was proposed, which are discussed in the next chapter.

6

STRUCTURAL SIMILARITY MEASURE

The great development of Semantic Web in the distributed environment leads to the different forms of ontologies. Therefore, ontology matching is an important task in order to share knowledge among applications more easily. In this chapter, a new structural measure is proposed to match ontologies. The I^3CON 2004 benchmark tests as well as Precision, Recall, and F-measure are used to evaluate our method.

The remainder of this chapter is organized as follows. Section 6.1 briefly overviews our approach. A description of our measure is given in section 6.2. In section 6.3 an example is provided in order to illustrate our structure-based method. In section 6.4 the experimental results of our method are described. Some conclusions are then reviewed in section 6.5. Finally, the chapter concludes with a short summary in section 6.6.

6.1. INTRODUCTION

The speedy development of the web technology leads to an increase in knowledge sharing among web applications. However, it is difficult to communicate between applications because these applications use different tools and knowl-

edge in a distributed system. Therefore, ontologies have been developed to express knowledge bases improving the understanding between applications. Nowadays ontologies can be utilized to represent and store knowledge in many different application domains such as peer-to-peer information sharing, information integration, e-commerce, web service composition; there are also a number of ontologies within the same subject. In fact, such ontologies are about the same area but they may use different concepts depended on background of knowledge, classification scheme, language employed because they are developed by different communities independently. Therefore, there is no ontology matched to another one perfectly. That is a reason why ontology matching based on the structures in the hierarchy of ontologies is necessary.

In this chapter, a edit-distance measure is used, which is a basic element-level technique and a new structure similarity measure to find correspondences between the concepts of source and target ontologies. In particular, Levenshtein measure is implemented to begin ontology matching processing. Firstly, Levenshtein measure is applied to each pair of concepts of the ontologies to obtain a lexical similarity matrix representing their similarity. Secondly, the lexical matrix is considered as an initial value for our structural similarity matrix, which is based on a new similarity measure. Next, by using a threshold value, a set of matched pairs of concepts is obtained. Finally, our structure-based ontology matching technique is improved by using centroid concepts which were proposed in [139].

6.2. OUR STRUCTURAL MEASURE

In this section, a new structure-based similarity measure for matching two nodes of the given ontologies based on combining lexical and structural similarities is presented. Firstly, two input ontologies are matched by using a basic lexical similarity measure (edit-distance) to obtain initial mappings. Based on that, our structure-based similarity measure is applied to find correspondences among the concepts of the ontologies.

In Fig. 6.1 the input ontologies are described as graphs. The nodes in the graphs correspond to entities in ontologies and the edges represent relationships between nodes which are connected. The process of similarity calculation gives values between all pairs of concepts c_i and c_j, where c_i and c_j belong to the ontologies O_1 and O_2, respectively. All these values are stored in a matrix. In this step an edit-distance similarity measure is used. Next, a new structure-based

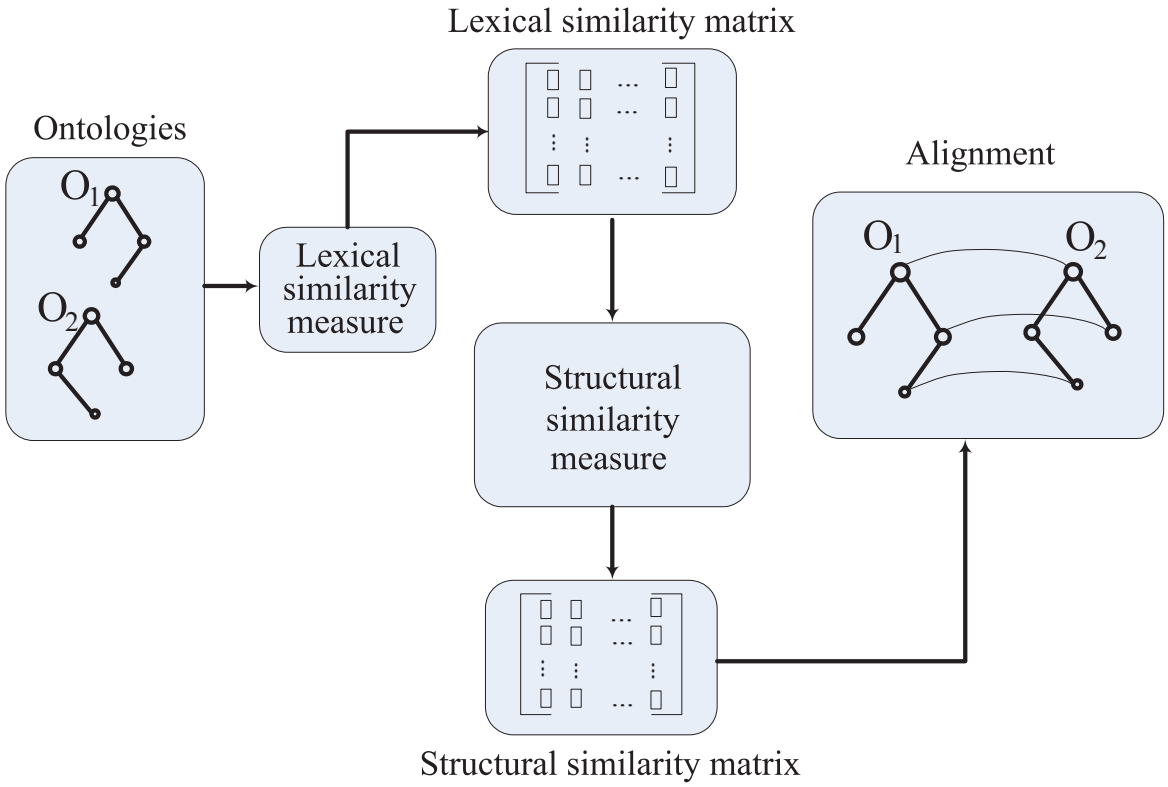

Figure 6.1: A new structural measure

similarity metric is applied for calculating similarity of each pair of nodes. At that time, the similarity degrees between each c_i in O_1 and all the nodes c_j in O_2 are obtained. Based on these results and a threshold *th*, the alignment is finally obtained. This structure-based method allows us to determine the similarity degree of two concepts based on the combinations of similarity measures of these concepts and their ancestors. The details of this calculation are explained in the subsequent subsections.

The lexical matching techniques focus on similarities of the entities of the given ontologies by computing string similarities of the entities. There have been many lexical similarity methods proposed so far. However, in this chapter only Levenshtein measure is considered to calculate the lexical similarity of two concepts.

As previously discussed in subsection 3.1.7, the Levenshtein measure [68] was employed to determine the number of differences between two strings. In particular, it computes the minimum number of operations needed to transform one string into another. Let c_i and c_j be two arbitrary strings. Three types of op-

erations are used including the substitution of a character of c_i by a character of c_j, the deletion of a character of c_i or the insertion of a character of c_j. The total cost of the operations used is equal to the sum of the costs of each of the operations.

The similarity measure for two strings (c_i, c_j) is defined as follows (see Section 3.1.7):

$$Lex\_sim(c_i, c_j) = \max\left(0, \frac{\min(|c_i|, |c_j|) - ed(c_i, c_j)}{\min(|c_i|, |c_j|)}\right) \tag{6.1}$$

where $|c_i|$, $|c_j|$ are the lengths of strings c_i and c_j, respectively; $ed(c_i, c_j)$ is the Levenshtein measure.

The following example is considered.

Given c_i="kitten" and c_j="kitchen" are two strings for which the similarity is computed. There are two edits transforming c_i into c_j:

- the substitution of "t" with "c"
- insert "h" follow "c"

Therefore, the Levenshtein distance between two strings "kitten" and "kitchen" is 2.

Applying the Eq. (6.1), the similarity between two strings "kitten" and "kitchen" is:

$$\begin{aligned} & Lex\_sim(kitten, kitchen) = \\ = \ & \max\left(0, \frac{\min(|kitten|, |kitchen|) - ed(kitten, kitchen)}{\min(|kitten|, |kitchen|)}\right) \\ = \ & \max\left(0, \frac{6-2}{6}\right) \\ = \ & 0.67 \end{aligned}$$

Now, Levenshtein measure is applied for computing the similarities between any two concepts with one from each ontology. The similarity matrix is obtained representing the lexical similarities of all pairs of nodes in the given ontologies. With this similarity matrix, the similarity between concepts based on our structural similarity measure is discussed in the following subsection.

Figure 6.2: The Goods ontologies

6.2.1. STRUCTURAL SIMILARITY

In this section, we propose *Struct_sim* metric for calculating the similarities between concepts of the input ontologies with the lexical similarity measure introduced in the previous section.

The two example ontologies are considered in Fig. 6.2.

Intuitively, the nodes Goods, Electronic_Products, Desktops and Printers from ontology O_1 are similar to the nodes Goods, Electronic_Products, Desktops and Printers from ontology O_2, respectively.

Our measure is based on the idea in [126] about the similarity between two concepts. However, the proposed method is different from the DSI (Descendant's Similarity Inheritance) method in [126]. In the DSI method, the authors concentrate on automatic structure-based technique to enhance the alignment results that were obtained from using the base similarity method, which is concept-based. Therefore, they point out not only the labels of the concepts but also the positions of the concepts in the hierarchy. The main characteristic of the DSI method is that it allows for the parent and in general for any ancestor to play a role in the identification of the concept [126]. It means the DSI method is based on a structure-based technique in which it uses the relations between ancestors. However, these relations are considered at the same levels. The idea of our new metric is expressed as follows: the similarity measure of any two concepts comes from the similarity of themselves and the contribution of their ancestors where their levels in the graphs can be different.

Relating the computations of the structural similarities, we can easily recog-

nize the important point with our intuitions that the more similar the structure is, the more likely nodes are similar. Besides, it can be assumed that two concepts are similar if their ancestors are similar even where the generations of ancestors can different.

Let m, n be the lengths of paths from the concepts c_i and c_j in the ontologies O_1 and O_2 to the roots, respectively (where $c_i \in O_1$ and $c_j \in O_2$); $Lex\_sim(c_i, c_j)$ stands for the lexical similarity of two concepts c_i and c_j; and $ancestor_k(c_i)$, $ancestor_l(c_j)$ for all the k^{th}, l^{th} concepts from the concepts c_i, c_j to the roots of the ontologies O_1 and O_2, respectively. The similarity between the concept c_i in the source ontology O_1 and the concept c_j in the target ontology O_2 is defined by the following equation:

$$Struct\_sim(c_i, c_j) = \alpha * Lex\_sim(c_i, c_j) + \beta * \frac{\left(\frac{\sum_{k=1}^{n} (n+1-k) * \max\{Lex\_sim(ancestor_k(c_i), ancestor_l(c_j)) | l=1..m\}}{n*(n+1)/2} + \frac{\sum_{l=1}^{m} (m+1-l) * \max\{Lex\_sim(ancestor_l(c_j), ancestor_k(c_i)) | k=1..n\}}{m*(m+1)/2} \right)}{2} \tag{6.2}$$

Therefore, our measure becomes:

$$Struct\_sim(c_i, c_j) = \alpha * Lex\_sim(c_i, c_j) + \beta * \left(\frac{\sum_{k=1}^{n} (n+1-k) * \max\{Lex\_sim(ancestor_k(c_i), ancestor_l(c_j)) | l=1..m\}}{n*(n+1)} + \frac{\sum_{l=1}^{m} (m+1-l) * \max\{Lex\_sim(ancestor_l(c_j), ancestor_k(c_i)) | k=1..n\}}{m*(m+1)} \right) \tag{6.3}$$

where the values of α and β satisfy the relationship: $\alpha + \beta = 1$.

By applying Eq. (6.3), when the algorithm finishes we obtain a structural similarity matrix using the lexical similarity matrix as an initial matrix.

In Eq. (6.3) our similarity measure takes values from the interval [0, 1] which can be found in the literature [37].

The main feature of the approach is that the similarity of two concepts in the source and target ontologies is not only the similarity of themselves but also the contribution of their ancestors. However, the closer the ancestor is to the root, the smaller role the ancestor has. The factors (n+1-k) and (m+1-l) perform this rule. Moreover, the measure is here applied for the maximum similarity of pairs of ancestors in order to contribute the similarities.

6.2.2. PROPERTIES OF OUR STRUCTURAL SIMILARITY MEASURE

Our structural similarity measure $Struct\_sim$ presented in Eq. (6.3) satisfies three properties of a similarity measure.

- Positiveness: $\forall c_i, c_j : Struct\_sim(c_i, c_j) \geq 0$

  Proof. Because $Lex\_sim(c_i, c_j), m, n, (n+1-k), (m+1-l), \alpha, \beta \geq 0$, $Struct\_sim(c_i, c_j) \geq 0$. □

- Maximality: $\forall c_i, c_j, c_t : Struct\_sim(c_t, c_t) \geq Struct\_sim(c_i, c_j)$

  Proof. $Lex\_sim(c_i, c_j) \leq 1 \Rightarrow Struct\_sim(c_i, c_j) \leq \alpha * 1 + \beta * 1 = \alpha + \beta = 1$. Besides, applying Eq. (6.3) leads to $Struct\_sim(c_t, c_t) = 1$. Therefore, $Struct\_sim(c_t, c_t) \geq Struct\_sim(c_i, c_j)$. □

- Symmetry: $\forall c_i, c_j : Struct\_sim(c_i, c_j) = Struct\_sim(c_j, c_i)$

  Proof. Because the concepts c_i and c_j have the same contribution in our structural measure and $Lex\_sim(c_i, c_j) = Lex\_sim(c_j, c_i)$, $Struct\_sim(c_i, c_j) = Struct\_sim(c_j, c_i)$ $\forall c_i, c_j$. □

6.2.3. IMPROVING THE STRUCTURE-BASED SIMILARITY MEASURE

In this study, the efficiency of the structure-based matching method by using a set of centroid concepts proposed in [139] is presented. In this approach, the authors chose a set of centroid concepts from two input ontologies to partition these ontologies. In particular, each concept in the ontologies is represented by its descriptive information including its name and comment. After that, the authors apply the element-level techniques and the vector space model method to achieve the similarities between names and comments, respectively. Finally, the overall similarity measures between concepts result from the combination of component similarities. In case one entity in an ontology matches perfectly to one in another ontology, these entities are picked as centroid concepts. The same set of centroid concepts is used as in [139]. In this chapter, the idea of using centroid concepts is that many concepts of the source ontology maybe appear in the target ontology in case these ontologies belong to the same domain [139] so

each centroid concept is considered as the anchor. This approach can reduce the computational complexity.

The structural similarities between nodes in ontologies as well as the ones of ancestors are calculated. The advantage of this improvement is that we do not need to consider all pairs of nodes from these nodes to roots. That means if there is any pair of centroid nodes on the path from the considered nodes to the roots, we only calculate the similarities of these nodes and their ancestors with the nearest pair of centroid concepts.

When applying Eq. (6.3), m and n are the lengths of the paths from c_i to the centroid node e_i of the ontology O_1 and c_j to the centroid node e_j of the ontology O_2, and $ancestor_k(c_i)$, $ancestor_l(c_j)$ for the k^{th}, l^{th} concepts from the concepts c_i, c_j to the centroid concepts e_i, e_j of the ontologies O_1 and O_2, respectively.

6.3. ILLUSTRATIVE EXAMPLE

In this section, an example illustrates our similarity matching method proposed in the previous subsection.

The two input ontologies are considered as shown in Fig. 6.2.

By applying the lexical matching method for all pairs of nodes in the source and target ontologies with one from each of the two ontologies, the similarity values between them are obtained. The result of the lexical measure is shown below.

$$Lex\_sim(O_1, O_2) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.06 & 0.11 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.13 & 0.14 & 0 & 0.11 & 0.38 \\ 0 & 0 & 0.13 & 1 & 0.43 & 0 & 0 & 0.25 \\ 0 & 0 & 0 & 0.29 & 0.14 & 0.14 & 0 & 0.14 \\ 0 & 0 & 0.38 & 0.25 & 0.14 & 0.14 & 0.13 & 1 \\ 0 & 0.07 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.14 & 0.43 & 1 & 0.29 & 0 & 0.14 \\ 0 & 0 & 0 & 0 & 0.29 & 1 & 0 & 0.14 \end{pmatrix}$$

At this time, the lexical similarity matrix is used to calculate the similarity values of all pairs of concepts based on their structures. Once applying the structure matching measure with Eq. (6.3), the similarities between concepts of the ontologies are obtained which are represented as follows:

Table 6.1: The matched pairs of concepts applying the threshold value $th = 0.7$.

No.	The matched concept in ontology O_1	The matched concept in ontology O_2
1	Goods	Goods
2	Electronic_Products	Electronic_Products
3	Computers	Computers
4	Desktops	Desktops
5	Printers	Printers
6	Laptops	Laptops
7	Tablets	Tablets

$$Struct\_sim(O_1, O_2) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0.23 & 0.20 & 0.20 & 0.20 & 0.23 & 0.20 \\ 0 & 0.27 & 0.42 & 0.26 & 0.26 & 0.26 & 0.35 & 0.26 \\ 0 & 0.20 & 0.92 & 0.28 & 0.29 & 0.19 & 0.34 & 0.43 \\ 0 & 0.19 & 0.32 & 0.95 & 0.58 & 0.30 & 0.23 & 0.40 \\ 0 & 0.20 & 0.27 & 0.38 & 0.29 & 0.29 & 0.27 & 0.27 \\ 0 & 0.19 & 0.48 & 0.32 & 0.25 & 0.25 & 0.32 & 0.79 \\ 0 & 0.28 & 0.35 & 0.26 & 0.26 & 0.26 & 0.35 & 0.26 \\ 0 & 0.20 & 0.36 & 0.46 & 0.83 & 0.37 & 0.27 & 0.27 \\ 0 & 0.20 & 0.27 & 0.18 & 0.37 & 0.83 & 0.27 & 0.27 \end{pmatrix}$$

In case the threshold value th is chosen to be 0.7, the matched pairs of concepts are obtained and given in Table 6.1.

By applying the centroid concepts algorithm, a set of centroid concepts from both ontologies including Goods and Electronic_Products is selected. Since Electronic_Products node is on the paths from all nodes to roots, we only need to calculate similarities of pairs of nodes and their ancestors to the node Electronic_Products. As a result, omitting the roots "Goods" in the matching algorithm leads to reducing the computation time.

6.4. EXPERIMENTS AND RESULTS

To test the performance of our structural similarity measure, five pairs of ontologies taken from the I^3CON 2004 data set are used, which are: People and pets

Table 6.2: F-measure values of DSI method and our measure for four pairs of ontologies.

Ontologies	F-measures	
	DSI	Struct_sim
People and pets	0.63	0.68
Weapons	0.87	0.95
Networks	0.45	0.49
Russia	0.57	0.57

(without instances), Weapons, Networks, Russia, and CS. The parameter α in Eq. (6.3) is also chosen as in [82]. The reason to choose these tests is that ontologies belonging to these tests are different about structures which emphasize the features of structural methods.

Because the DSI method did not implement the pair of ontologies CS, our similarity measure was compared with the DSI method based on four pairs, which are People and pets (without instances), Weapons, Networks, and Russia. A comparison of our results with Avg. F-measures which are the F-measure average values of six participants executed five pairs of ontologies (People and pets (without instances), Weapons, Networks, Russia, and CS) in the I^3CON 2004 benchmark is also given. These participants include a research program of Lockheed Martin ATL, Intelligent Agent Systems from AT&T, Institut AIFB from University of Karlsruhe, an algorithm from INRIA, Lexicon-based Ontology Mapping (LOM) from Teknowledge [52], and Similarity Flooding algorithm [74]. Note that no participant of these participants except Similarity Flooding algorithm implements the pair of ontologies Weapons.

The implementation of the structural similarity was evaluated by using the following classical measures: Precision, Recall and F-measure.

Table 6.2 shows F-measure values of DSI method and our measure for four pairs of ontologies. As can be seen from the results described in Table 6.2, our measure has a better F-measure value than original DSI method generally. It means it is more effective than the DSI method. Similarly to the DSI approach, our measure depends on initial similarities between pairs of nodes. Moreover, both our measure and the DSI method are based on the similarities of their descendants. However, the DSI method only is based on the computation of similarities of ancestors with the same levels.

As we know ontologies normally have differences in details. That means par-

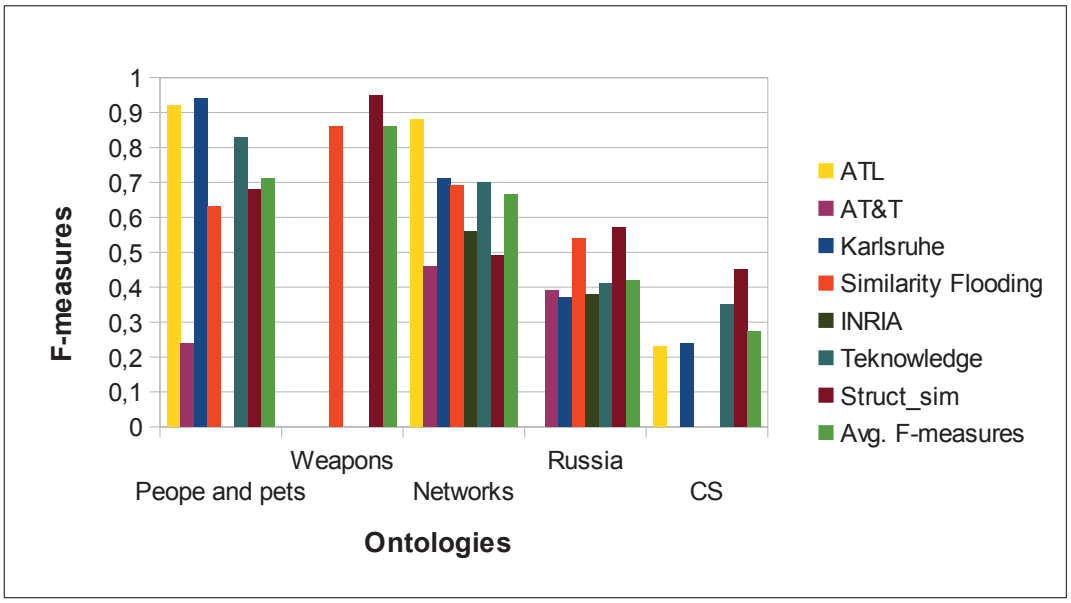

Figure 6.3: Approaches vs. F-measures

ent of a node in one ontology may be not similar to the parent of the corresponding node in the other ontology, but to a node at a higher level. This can lead to the situation that the DSI method omits some ancestors in one ontology if they do not have counterparts at the same level in the other ontology. By not finding the corresponding ancestors the similarity values will be significantly smaller. On the other hand, if ontologies have the same structural similarities, our approach and the DSI method are coincident. As a result, the F-measure values obtained from two approaches are similar.

In order to compare the performance of our measure with the other ones, F-measure values applying our method, Similarity Flooding algorithm, and measures of five participants based on five pairs of ontologies [5] were conducted and shown in Fig. 6.3. Note that no one of these methods is better than ours for all pairs of ontologies. For the test case People and pets, the F-measure value of our measure is less than the average F-measure value of six participants. However, our result is better than these of two others - the AT&T and Similarity Flooding methods. The test case Weapons has good structural and lexical characteristics so applying our measure yields a good result compared to Similarity Flooding method and the average F-measure. Since the labels of the nodes in the test case Networks contain some omitted words, our method is not so good in compari-

[5] http://www.atl.external.lmco.com/projects/ontology/papers/I3CON-Results.pdf

son with the Similarity Flooding method as well the average F-measure. Matching pairs of ontologies CS and Russia is difficult because these ontologies have big differences in their structures. In particular, the CS ontologies consist of 109 concepts and 52 edges, 20 concepts and 7 edges respectively, while the Russia ontologies contain internal structures and the labels of the concepts are very different [29], which leads to not good results of the approaches. For the CS ontologies, all F-measure values obtained by participants and our approach are less than 0.5. However, our method outperforms the other methods including ATL, Karlsruhe, and Teknowledge. Particularly, F-measures of these methods are 0.45, 0.23, 0.24, and 0.35, respectively. The test case Russia includes a lot of properties and instances so a good result is not achieved. However, even in this case, the F-measure applying our method is the best value.

6.5. CONCLUSIONS

In this chapter, a new measure was proposed to find correspondences of concepts of two input ontologies based on the combination of lexical and structural similarities. A basic lexical measure was used for computing the lexical similarities among nodes. These similarities are used as initial values for estimating structural similarities. The structure-based similarity value of two nodes is the contribution of the lexical similarity of the concept names and similarities of their descendants. With the proposed similarity measure, a structural matrix describing the similarities of all pairs of nodes is received.

The other methods based on structure usually focus on the similarities of neighbor nodes such as parents, grandparents, children and siblings (see more in Section 3.3). The important aspect of our structural approach is that the similarity of a pair of nodes depends not only on their similarity but also on the similarity of all possible pairs of their ancestors. Unlike the approach in [126], our method does not omit any pairs in order to deal with the situation that the ontologies do not have the same structure.

For improving the structural matching, the similarities between two nodes based on similarities of their descendant to the nearest centroid nodes instead of roots are calculated. This implementation leads to reducing the calculation.

Our metric was conducted on the I^3CON 2004 [119] data set. Our similarity measure was compared to the DSI method and Similarity Flooding algorithm as well as measures of five participants. The experimental results showed that our approach possesses some prominent features comparing to ontology matching

algorithms and is effective in case the given ontologies are very different in their structures.

In the future research, our approach should be integrated with linguistic matching using WordNet dictionary in order to increase the accuracy of the lexical matching and to match the relations of the given ontologies. With this method ontologies can be matched based on weighted graphs using the properties as the weights.

6.6. SUMMARY

In this chapter, an automatic ontology matching method was proposed by combining lexical and structure-based measures. A basic lexical similarity measure was applied to all pairs of concepts of two ontologies to achieve an initial matrix. With this matrix, the similarities between concepts are calculated based on a new structural similarity measure. Additionally, the structure-based matching method was improved by using a set of centroid concepts to reduce the computation time. An illustrative example was shown to express our idea more clearly. I^3CON 2004 benchmark is used to evaluate the proposed method. The experimental results showed that our measure has some prominent features for ontology matching.

The three novel similarity matchers described in previous chapters need to be integrated. For this purpose, a combination of our proposed methods including the element, structural, and semantic approaches is discussed in the next chapter of the thesis in detail.

7

Integrated Ontology Matching and Evaluation of Our System

This thesis has four main contributions including the proposed lexical, structure, semantic similarities, and the combination of these measures. In the previous chapters, the single similarities are discussed separately. The first contribution showed in section 4.2 is based on combining information-theoretic and edit distance measures. The second one described in section 5.2 indicates the semantic similarities among entities by using the WordNet dictionary. The third contribution explained in section 6.2 calculates the structural similarity degrees of entities in the hierarchy. In this chapter, the fourth one based on our approach in [87] gives a presentation of how measures are combined for the overall ontology matching task. The benchmark tests of the 2008 OAEI and the classical measures including Precision, Recall, and F-measure are used to implement and evaluate our integrated approach.

The remainder of this chapter is organized as follows. We start by introducing our approach in general in section 7.1. For each measure (e.g. lexical, semantic, and structure similarities), we obtain a similarity value matrix. These results are then integrated together to yield the overall similarity. We use a weighted sum method to combine these measures in which a weight is assigned to each component. In section 7.2, the proposed ontology matching framework and a detailed description of our approach are provided. Additionally, a discussion and evalu-

ation of the results are presented in section 7.3. A conclusion of our results and future work are pointed out in section 7.4. Finally, section 7.5 gives a summary to close this chapter.

7.1. INTRODUCTION

Many ontology matching systems have been proposed so far based on lexicon, structures, instances, semantic, and combination of the above approaches (AnchorFlood [116], DSSim [79], MapPSO [14], TaxoMap [46], GLUE [27], iMAP [21], AROMA [19], NOM [33], QOM [31], SAMBO [66]). This chapter takes into account the combination of different matching strategies including lexical-based, structure-based, and semantic-based methods to obtain a final alignment. In particular, our approach focuses on names, labels, comments, positions of concepts in the hierarchy, relationships between these concepts, and semantics based on WordNet. However, our approach does not consider instances data and user's feedback. Our matching process uses sequential and parallel strategies in which the sequential phase is based on combining lexical and structural measures and the parallel phase is relied on combining semantic measure and structural similarity values obtained in the previous step.

7.2. ARCHITECTURE

In this section, a framework for automatic ontology matching is described. Ontology matching is divided into two main strategies including the sequential and parallel compositions [37] to obtain alignments between input ontologies. Our framework supports some matching approaches and also applies both strategies. Fig. 7.1 shows the two phases in our framework. For the sequential phase, the lexical similarity values are applied to structural method to create a similarity matrix while the parallel composition phase is the combination of structure-based and semantic-based measures. The processes of similarity calculation return values between all pairs of concepts in two ontologies. All these values are stored in the structure and semantic matrices, respectively. Each pair of concepts from these two matrices is combined by using weights, then the overall similarity values are produced. Based on these results and a threshold *th*, the alignment is

finally obtained. The similarity between two entities in the given ontologies depends on the similarities of their components and structures. In this study, the

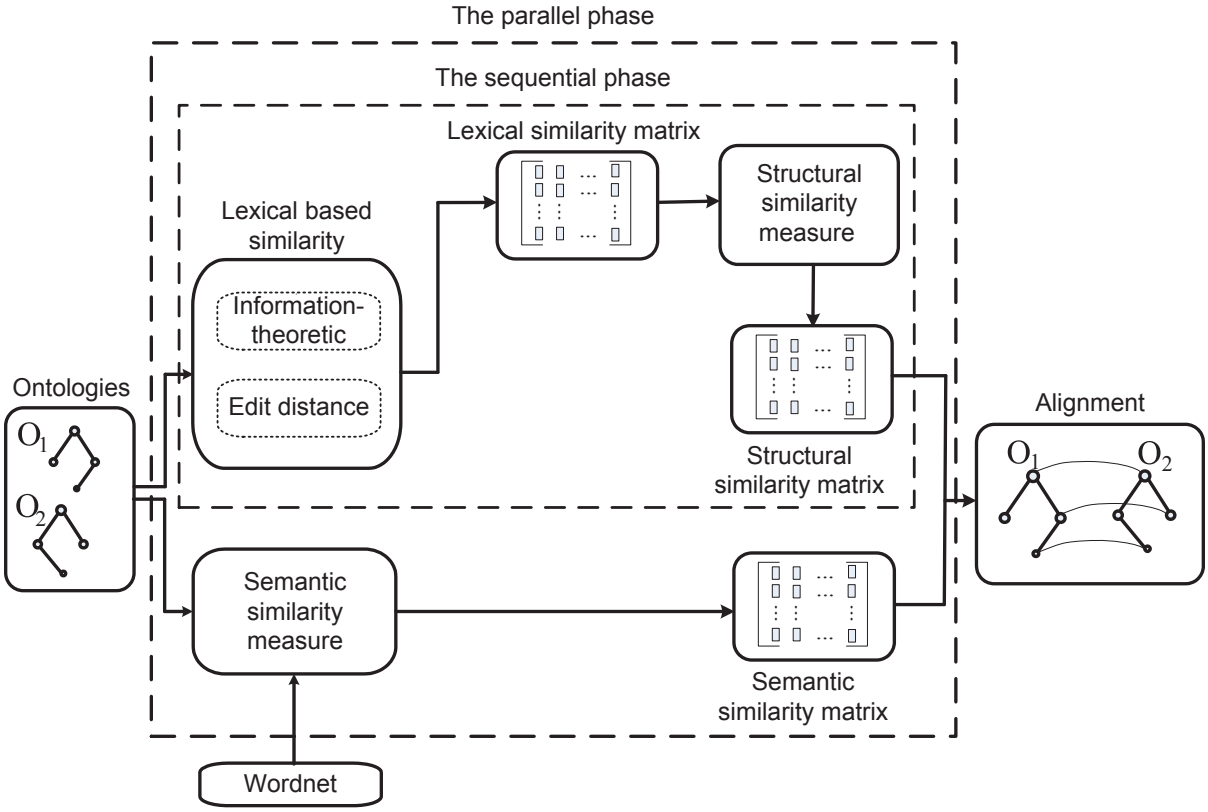

Figure 7.1: Framework for ontology matching

considered components including names, labels, comments as well as relations and structures among entities in two ontologies are taken to calculate the similarity among these entities. There are many techniques to aggregate similarities, for example weighted product, weighted sum, weighted average, fuzzy aggregation, voting, and arguing [26, 37]. Thanks to parameters, various matching systems are composed by a set of individual measures to produce good alignments in an optimal and flexible way [27, 31, 50, 58, 66, 72]. In fact, each ontology has its own characteristic. Therefore, depending on the features of ontologies and application domains are chosen these parameters should be changed. Similar to AgreementMaker's [18] and ASCO's [7] systems, the component and the combined similarity results in our work are computed by using weighted average and weighted sum methods in case they have more than one similarity degree, respectively. The details of our approach are explained in the subsequent sections.

7.2.1. RELATED DEFINITIONS

Let O_1 and O_2 be two ontologies, entities belonging to these ontologies are e_1 and e_2, respectively. Entities usually consist of their names, denoted as $name(e_1)$ and $name(e_2)$, their labels, denoted as $label(e_1)$ and $label(e_2)$, and their comments, denoted as $comm(e_1)$ and $comm(e_2)$. The overall similarity value between two entities e_1 and e_2 is defined as $Overall\_Sim(e_1, e_2)$. This value results from the fundamental similarities obtained in two following phases.

- The sequential phase: in this phase, the structural similarities depend on the lexical similarities calculated in the previous step and the positions of entities in ontologies. The structural similarity between entities is defined as $Struct\_sim(e_1, e_2)$. The lexical similarity (also called string-based similarity), $Lex\_sim(e_1, e_2)$, comes from cooperation between information-theoretic and edit distance approaches.

- The parallel phase: the result of this phase is the overall similarity integrated from structural and semantic measures multiplied by weights. The semantic similarity between entities (also called knowledge-based similarity), $Sem\_sim(e_1, e_2)$, is determined by relationships, semantics, and structures of these entities in hierarchy of WordNet.

Both lexical and semantic similarity degrees depend on three component similarities including class names, labels, and comments of entities. In general, by assigning a weight to each of the component similarity, lexical and semantic similarities are described as follows:

$$Sim(e_1, e_2) = \frac{\sum_{k=1}^{3} w_k * sim_k(e_1, e_2)}{\sum_{k=1}^{3} w_k} \tag{7.1}$$

where w_k are weights corresponding to features, $sim_k(e_1, e_2)$ are component similarities.

If two entities e_1 and e_2 do not contain a feature (for example, comments), the similarity of that feature is ignored. In this case, its corresponding weight w_k is assigned to 0. If one feature belongs to only one entity, its corresponding weight is set to 0 and then the similarity between two entities is defined as

$$Sim(e_1, e_2) = \max(Sim(e_1, e_2) - \lambda, 0) \tag{7.2}$$

The Eq.(7.2) is based on the reasons as follows. The first reason is that, if two concepts have the same features and the component similarity values between these concepts equal 1, these concepts are chosen as centroid concepts and will be used for calculating the structural similarity values. On the other hand, these concepts may not match perfectly. Secondly, according to our intuition, the similarity degree of two concepts is based on various features such as class names, labels, comments, and so on. For non-existence feature, the similarity value $Sim(e_1, e_2)$ between two concepts should be reduced by λ. The λ value should large enough so that concepts have sufficient difference. Therefore, λ value here is chosen equal to 0.05. Moreover, maximum function is applied to yield non-negative similarity values.

7.2.2. MEASURING STRUCTURAL SIMILARITY

At the first time, when the lexical measure is used, the similarity for each component of each pair of entities in two ontologies is obtained. After getting the lexical similarity values between entities, the combination of lexical-based and structure-based metrics together is implemented.

LEXICAL-BASED SIMILARITY

Lexical-based method is separately applied to names, labels, and comments of entities in two ontologies to achieve the similarities of each component of these entities.

- The similarities of class names and labels: normally, class names and labels are text chains such as words, the combination of a few words together without blank spaces, so they are short. The lexical similarity measure we proposed in [85] was applied for calculating the similarities of these class names and labels.

$$Lex\_sim(e_1, e_2) = \tag{7.3}$$
$$= \frac{\alpha(\max(|e_1|, |e_2|) - ed(e_1, e_2))}{\alpha(\max(|e_1|, |e_2|) - ed(e_1, e_2)) + \beta(|e_1| + |e_2| - 2\max(|e_1|, |e_2|) + 2ed(e_1, e_2))}$$

  where $ed(e_1, e_2)$ is Levenshtein measure. Let us consider the following example.

Example. Given names of two entities:
$name(e_1)$="Proceedings" and $name(e_2)$="InProceedings".

The Levenshtein distance between these strings is 2. In addition,
$|Proceedings| = 11$, $|InProceedings| = 13$,
$\max(|Proceedings|, |InProceedings|) = 13$.
By applying Eq.7.3, the similarity between two strings
"Proceedings" and "InProceedings" is:
$Lex\_name(Proceedings, InProceedings) = 0.733$

- The similarities of comments: classes usually contain comments describing these classes. However, comments are usually short texts too. To determine the similarity between two comments, two steps including normalization and comparison steps were executed. In the normalization step, we broke each comment into the ordered sets of tokens in which tokens are in the order of their appearance in the comment and then removed stopwords (for example, *the, a, and, of, to*), blank spaces, punctuation, symbols, replaces abbreviations (for example, *PC* → *Personal Computer*, *OS* → *Operating System*), and so on. Let $Comm1$ and $Comm2$ are two ordered sets of tokens of comments of two entities e_1 and e_2 in input ontologies O_1 and O_2, respectively. $Comm1$ and $Comm2$ can be presented as

  $Comm1 = \{\text{comm}(e_1)_1, \text{comm}(e_1)_2, ..., \text{comm}(e_1)_n\}$,

  $Comm2 = \{\text{comm}(e_2)_1, \text{comm}(e_2)_2, ..., \text{comm}(e_2)_m\}$.

  In the comparison step, these similarities are calculated in the same way as the similarities of class names and labels but applied to tokens. In particular, after converting comments $Comm1$ and $Comm2$ to the ordered sets of tokens, every token will be considered as a unit. At that time, a set of tokens in each comment will be linked together as a string of units. The Levenshtein measure [68] is used to calculate distance between these strings. We will illustrate this idea with the following example.

Example. Given comments of two classes:
$comm(Proceedings)$="The proceedings of a conference." and
$comm(InProceedings)$="An article in a conference proceedings.".
The sets of ordered tokens of comments are:
$Comm1$={proceedings, conference} and
$Comm2$={article, conference, proceedings}.
Therefore, the Levenshtein distance between two comments
$comm(Proceedings)$ and $comm(InProceedings)$ is 2.
In addition, $|comm(Proceedings)| = 2$, $|comm(InProceedings)| = 3$,

$\max(|comm(Proceedings)|, |comm(InProceedings)|) = 3$.
Applying Eq.(7.3), the similarity between these two comments is:

$$\begin{aligned} & Lex\_comm(comm(Proceedings), comm(InProceedings)) = \\ = & \frac{\alpha * (\max(2,3) - 2)}{\alpha * (\max(2,3) - 2) + \beta * (2 + 3 - 2 * \max(2,3) + 2 * 2)} \\ = & \frac{0.2}{0.2 + 3 * 0.4} = 0.143 \end{aligned}$$

In our approach, each concept in the ontologies is represented by its descriptive information including its name, label, and comment. Applying the lexical similarity measure achieves the similarities between names, labels, and comments, respectively. After calculating lexical similarities between each concept in source ontology to all concepts in target ontology, three similarity matrices of classes, labels, and comments are obtained. By applying Eq.(7.1), the lexical similarity between e_1 and e_2 is presented as

$$\begin{aligned} Lex\_sim(e_1, e_2) = & \frac{w_n * Lex\_name(e_1, e_2)}{w_n + w_l + w_c} \\ & + \frac{w_l * Lex\_label(e_1, e_2)}{w_n + w_l + w_c} \\ & + \frac{w_c * Lex\_comm(e_1, e_2)}{w_n + w_l + w_c} \end{aligned} \tag{7.4}$$

where w_n, w_l, w_c, $Lex\_name(e_1, e_2)$, $Lex\_label(e_1, e_2)$, and $Lex\_comm(e_1, e_2)$ are weights and component similarities corresponding to features class names, labels, and comments, respectively.

The string-based measure shown in Eq. (7.4) is used for computing the similarity matrix representing lexical similarities between any two concepts with one from each ontology. This matrix is also employed to compute the similarity values of all pairs of concepts in ontologies based on the structure-based measure as discussed in the following subsection.

STRUCTURE-BASED METHOD

In this phase, a structure-based similarity metric we proposed in [83] was applied for calculating similarity of each pair of concepts. The initial matrix is the lexical similarity matrix introduced in the previous subsection.

In case each entity in an ontology matches perfectly to one in another ontology ($Lex\_sim(e_1, e_2) = 1$), these entities are picked as centroid concepts. At that time, a set of centroid concepts is obtained.

The process of similarity calculation gives values between all pairs of concepts between e_1 and e_2, where e_1 and e_2 belong to the ontologies O_1 and O_2, respectively. All these similarity values are then stored in a structural matrix.

In fact, the structure of entities refers to how an entity is related to others. In addition, the structure contains a lot of the semantics of the entities that they express as well as the similarity degree value between two arbitrary entities. Therefore, semantic measure described hereafter will be integrated with our structural technique together.

7.2.3. SEMANTIC SIMILARITY MEASURE

WordNet is considered as a background knowledge source to take semantics of terms. In this section, our measure proposed in [84] and the method in [7] were applied to calculate the semantic similarity. Class names, labels, and comments are conventionalized to sets by tokenizing them based on upper case, punctuation, symbols, and so on. Each token in a set (for example, comments) is compared with all tokens from the same type of set (any two tokens with one from each of set), and then the best similarities are chosen. The average of all best similarities of the same type is the semantic similarity between two objects. For example, the semantic similarity between two comments is described as:

$$Sem\_comm = \frac{\sum_{i=1}^{n} \max(comm(e_1)_i, Comm2) + \sum_{j=1}^{m} \max(comm(e_2)_j, Comm1)}{n+m} \quad (7.5)$$

where n and m are the numbers of tokens in the sets of comments $Comm1$ and $Comm2$, respectively.

Example. Using the two entities from the previous section:
$name(e_1)$="Proceedings" and $name(e_2)$="InProceedings".
The similarity between the two strings "Proceedings" and "InProceedings" is:
$Sem\_name(Proceedings, InProceedings) = 0.767$

Example. Given comments of two entities:
$comm(Proceedings)$="The proceedings of a conference." and
$comm(InProceedings)$="An article in a conference proceedings.".

The sets of ordered tokens of comments are
$Comm1$={proceedings, conference} and
$Comm2$={article, conference, proceedings}, respectively.
The similarity between these two comments is:
$Sem\_comm(Proceedings, InProceedings) = 0.870$
Semantic similarities between concepts result from the combination of component similarities.

$$Sem\_sim(e_1,e_2) = \frac{w_n * Sem\_name(e_1,e_2)}{w_n + w_l + w_c} + \frac{w_l * Sem\_label(e_1,e_2)}{w_n + w_l + w_c} + \frac{w_c * Sem\_comm(e_1,e_2)}{w_n + w_l + w_c} \tag{7.6}$$

where $Sem\_name(e_1,e_2)$, $Sem\_label(e_1,e_2)$, $Sem\_comm(e_1,e_2)$ are the semantic similarities of names, labels, and comments, respectively.

7.2.4. COMBINING SIMILARITY VALUES

The similarities are combined to get an overall similarity matrix representing the similarities of every pair of entities in given ontologies.

$$Overall\_Sim(e_1,e_2) = w_1 * Struct\_sim(e_1,e_2) + w_2 * Sem\_sim(e_1,e_2) \tag{7.7}$$

where $\sum_{t=1}^{2} w_t = 1$

In case the final similarity of two entities is equal or higher than the threshold, these entities are considered similar. Consequently, one entity in an ontology can be similar to some entities in the other. It means, our system can output one-to-one and one-to-many alignments.

7.3. EVALUATION

The datasets were taken from OAEI benchmark 2008 to test and evaluate the performance of our system and other ones. Ontologies in this benchmark test were modified from the reference ontology 101 and can be divided into three categories: 101-104 (1xx), 201-266 (2xx), and 301-304 (3xx). Besides, ontologies 301-304 present real-life ontologies for bibliographic references found on

the web. Since ontology 102 focus on wine which is irrelevant for the domain of bibliography, it is ignored. Ontology matching systems are chosen to compare including CIDER [44], Spider [112], GeRoMe [60], Anchor-Flood [116], Lily [137], DSSim [79], MapPSO [14], TaxoMap [46], MLMA+ [2], Akbari&Fathian [1], and ours (called LSSOM - Lexical Structural Semantic-based Ontology Matching method). The implementation of these approaches was evaluated based on the classical measures including Precision, Recall, and F-measure.

In our experimentation, the weights corresponding to features (class names, labels, and comments) and the partial similarity values (structural and semantic similarities) are assigned the fixed values 0.5, 0.25, 0.25, 0.5, and 0.5, respectively). Moreover, according to a suggestion in Chapter 4, parameters α and β in are in range from 0.2 and 0.35, from 0.4 and 0.325, respectively. Here, these parameters $\alpha = 0.25$ and $\beta = 0.375$ are chosen. The Table 7.1 shows the average Precision, Recall, and F-measure values of categories and all ontologies in this benchmark test.

In the benchmark, ontologies in groups 1xx have good information, for instance, class names, labels, comments, and structures in the hierarchy. As a result, Precision and Recall values of all the systems are quite high, except that Recall of GeRoMe and MLMA+ are not good and Recall of TaxoMap is very low (0.34). Our approach and other systems (for example, Lily, Anchor-Flood, and DSSim) give Precision and Recall values of 1. Consequently, F-measure values are also equal to 1.

In the tests 2xx, ontologies miss some features from the reference ontology. The tests 2xx include of three main groups: 201-210, 221-247, and 248-266. For tests 201-210, class names are arbitrary strings while some other information is lost such as labels and comments. In tests 221-247, the structures of the ontologies can be either cut down or expanded in term of size. However, systems using structural technique also introduced good results even similar to the tests 1xx. Of course, Precision and Recall values of all the ontology matching systems are slightly worse than those for tests 101-104. The tests 248-266 have not good class names and structures, so the quality of the matchers is not good. As can be seen in the Table 7.1, Precision values in the tests 2xx are either higher than 0.90 or less than 0.6 while Recall values are quite low in general. There are only three systems having good values (Lily, Akbari&Fathian, and LSSOM).

For the real-world tests 301-304, Precision and Recall values are changed in the range between 0.15 (Spider) and 0.95 (Anchor-Flood) for Precision values and 0.21 (TaxoMap) and 0.84 (Akbari&Fathian) for Recall values.

Table 7.1: Average Precision, Recall, and F-measure values of different approaches for three categories of ontologies in the benchmark OAEI 2008 (Pre.=Precision, Rec.=Recall).

Approaches		101-104	201-266	301-304	Average	F-measure
CIDER	Pre.	0.99	0.97	0.90	0.97	0.76
	Rec.	0.99	0.57	0.75	0.62	
Spider	Pre.	0.99	0.97	0.15	0.81	0.71
	Rec.	0.99	0.57	0.81	0.63	
GeRoMe	Pre.	0.96	0.56	0.61	0.60	0.59
	Rec.	0.79	0.52	0.40	0.58	
Anchor-Flood	Pre.	1.0	0.96	0.95	0.97	0.82
	Rec.	1.0	0.69	0.66	0.71	
Lily	Pre.	1.0	0.97	0.87	0.97	0.92
	Rec.	1.0	0.86	0.81	0.88	
DSSim	Pre.	1.0	0.97	0.90	0.97	0.79
	Rec.	1.0	0.64	0.71	0.67	
MapPSO	Pre.	0.92	0.48	0.49	0.51	0.52
	Rec.	1.0	0.53	0.25	0.54	
TaxoMap	Pre.	1.0	0.95	0.92	0.91	0.35
	Rec.	0.34	0.21	0.21	0.22	
MLMA+	Pre.	0.91	0.57	0.68	0.69	0.67
	Rec.	0.89	0.52	0.65	0.65	
Akbari&Fathian	Pre.	0.98	0.78	0.87	0.86	0.84
	Rec.	0.95	0.74	0.84	0.83	
LSSOM	Pre.	1.0	0.90	0.98	0.96	0.87
	Rec.	1.0	0.72	0.74	0.80	

In short, although average Precision value of TaxoMap system is high, its average F-measure value is the worst because its Recall value is also the worst. The MapPSO system is better than TaxoMap about the average F-measure, but it does not bring a good value. Anchor-Flood, Akbari& Fathian, and LSSOM approaches return average F-measure quite high: 0.82, 0.84, and 0.87, respectively. Lily is still considered the best ontology matching system. However, this system uses instances in matching. Our approach has not use instances yet, which is different from Lily system. Our approach is highly significant compared to the other ontology matching systems which do not use instances data. In addition, it is considered as one of the best ontology matchers on the OAEI 2008 benchmark

test.

7.4. CONCLUSIONS

This chapter presented an approach to generate correspondences among entities of two input ontologies based on lexical-based, structure-based, and semantic-based measures in detail. In this work, our system implements two phases which are sequential and parallel strategies. In the sequential phase, a structural similarity matrix applied the structure-based metric is produced by the subsequent lexical-based measure. Thanks to the weighted sum method, the combination of structural and semantic matchers in the parallel phase, and a certain threshold as well gives the final alignment. Consequently, our approach can induce one-to-one and one-to-many alignments. In addition, the results of our approach in the benchmark dataset of the 2008 OAEI were described. The experimental results demonstrate that our approach which automatically matches without instances achieves the high F-measure values.

Instances information of ontologies will be integrated in our approach in order to increase the accuracy of the final alignment. Moreover, machine learning techniques should be used to obtain a better quality of matching results. Our approach should also be tested on larger ontologies, evaluate its performance, and efficiency in the future work.

7.5. SUMMARY

This chapter presented an ontology matching approach which brings a final alignment by combining three kinds of different similarity measures: lexical-based, structure-based, and semantic-based techniques as well as using information in ontologies including names, labels, comments, relations and positions of concepts in the hierarchy and integrating WordNet dictionary. Firstly, two ontologies were matched sequentially by using the lexical-based and structure-based similarity measures to find structural correspondences among the concepts. Secondly, the semantic similarity based on WordNet dictionary was applied to these concepts in given ontologies. After the semantic and structural similarities were obtained, they were combined in the parallel phase to yield the final similarities. Our system was implemented and evaluated based on the OAEI 2008 benchmark dataset. The experimental results showed that our approach obtains good F-measure values and outperforms other automatic ontology matching systems

which do not use instances information.

In the next chapter, a summary with the contributions and some suggestions for future directions are given to conclude this thesis.

8

Conclusions and Future Works

This final chapter summarizes the main contributions and future works of our research about ontology matching in Semantic Web. In section 8.1, the significant contributions to find correspondences of the entities of two given ontologies and the advantages compared to similar approaches are discussed. Section 8.2 closes the thesis by pointing out some possible improvements and bringing future directions of our approaches.

8.1. Conclusions

Ontologies have become an important part of a variety of application fields. Many of the existing ontologies are created and developed with similar purposes by different research communities and are widely distributed. Therefore, they can contain different terms, structures, and levels of detail (also called ontology heterogeneity). To deal with ontology heterogeneity, many research groups concentrate on developing ontology matching systems, based on different techniques of similarities such as semantic, syntactic, terminological, structural, and extensional. In the current section, we will outline a set of the research work that has been described in the previous chapters.

In this thesis, the definitions, applications, and the need for ontology matching were also introduced. This work also collected and analyzed the techniques of ontology matching, similarity measures and benchmark data sets. The research presented in this thesis also consists of related matching works based on lexical, semantic and structural techniques. Our approaches consist of structure-based, lexical-based, semantic-based measures and a combination of these methods for ontology matching. In the semantic similarity degree solution, the WordNet dictionary was used to find the matching pairs based on synonyms and their relationships. A measure combined by the information-theoretic and edit distance methods for calculating the lexical similarity was performed. The structure of ontologies is important [37], so that it should be taken into account in order to find concept matchings among all possible pairs of concepts. For that reason, our study focused on the structures of entities in given ontologies. The single techniques were then integrated by using the weighted sum method to obtain the alignment. The alignments consist of one-to-one as well as one-to-many in which each entity from the source ontology returns one or many matched corresponding entities in the target ontology. Moreover, our measures and framework used well-known benchmark datatests to implement. The experimental sections of our work in chapters 4, 5, 6, and 7 contain evaluations, discussions, and comparisons between the proposed similarity approaches and others. These results indicated that our approach yields enhanced match results compared to others in terms of Precision, Recall, and F-measure.

Consequently, there are some remarkable differences between our approach and others. The main advantages of our methods compared to state of the art approaches are a result of the following properties.

- In our lexical method, the similarity of two entities is a combination of information-theoretic and edit distance methods. It depends on both common and different features of these entities. In addition to that, the similarity of the strings is also based on the number of operations required to transform each of string into other one.

- The semantic measure takes account of hypernym/hyponym relationships between the considered concepts in the WordNet. The important features of this measure are that it considers the relevancy links between concepts and applies the edge-counting based method to decide the semantic similarities.

- The structural method does not only depend on the positions of the con-

cepts but also the relationships between their ancestors whereby their levels in the hierarchy can be different. In order to the calculation time is decreased, the ancestors are chosen as a set of centroid concepts instead of the root node. Consequently, the overall running time is reduced.

- The suggested approaches can be applied on ontologies in various domains.

8.2. FUTURE WORKS

In addition to the contributions made by this thesis, the similarity measures, performance and efficiency of the matching approaches proposed should be improved. Some of the potential extensions in the future works are presented hereafter.

The proposed semantic similarity approach of the thesis is dedicated to hypernym/hyponym relationships among entities. However, other relations such as "part-of" should be considered to decide the relatedness between the pairs of terms. Furthermore, in order to increase the matching quality instances information of ontologies will be integrated in our approach.

In our study, the similarities between entities of two ontologies depend on a set of parameters. These parameters consist of the threshold value and the parameters assigned to the component similarities. The different values of each parameter will bring different similarities of entities. However, the selection of the optimal parameters which include the threshold, alpha, beta, and gamma constraints to pick the best match results of systems is usually not easy [81]. Moreover, for each kind of ontology these parameters can be assigned different values. In the future work, we plan to apply machine learning method in combination of single measures, which can overcome this limit. A further step would be to add a new method which employs clustering strategy.

In the implementation of system, our approach was only tested on the small ontologies benchmark datasets, so the main focus of our work in the near future will be on larger tests. Additionally, complexity of algorithms and efficiency of approaches will also be considered in the future work.

REFERENCES

[1] Ismail Akbari and Mohammad Fathian. A Novel Algorithm for Ontology Matching. *Information Science*, 36(3):324–334, 2010.

[2] Ismail Akbari, Mohammad Fathian, and Kambiz Badie. An Improved MLMA+ Algorithm and its Application in Ontology Matching. In *Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA)*, pages 56–60. IEEE, 2009.

[3] Ahmed Alasoud, Volker Haarslev, and Nematollaah Shiri. An Empirical Comparison of Ontology Matching Techniques. *Information Science*, 35(4):379–397, 2009.

[4] Alsayed Algergawy, Eike Schallehn, and Gunter Saake. A Sequence-based Ontology Matching Approach. In *The 10th International Conference on Information Integration and Web-based Applications & Services*, pages 131–136. ACM, 2008.

[5] Marco A. Alvarez and Lim SeungJin. A Graph Modeling of Semantic Similarity between Words. In *The International Conference on Semantic Computing*, pages 355–362. IEEE, 2007.

[6] Thanh Le Bach and Rose Dieng-Kuntz. Measuring Similarity of Elements in OWL DL Ontologies. In *The 1st International Workshop on Contexts and Ontologies (C&O) at the 20th National Conference on Artificial Intelligence (AAAI)*, pages 96–99, 2005.

[7] Thanh Le Bach, Rose Dieng-Kuntz, and Fabien Gandon. On Ontology Matching Problems for Building a Corporate Semantic Web in a Multi-Communities Organization. In *The 6th International Conference on Enterprise Information Systems*, pages 236–243, 2004.

[8] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. ACM press Addison Wesley, 1st edition, 1999.

[9] Sidney C. Bailin and Walt Truszkowski. Ontology Negotiation: How Agents can Really Get to Know Each Other. In Walt Truszkowski, Mike Hinchey, and Chris Rouff, editors, *Innovative Concepts for Agent-based Systems*, volume 2564 of *LNCS*, pages 320–334. Springer-Verlag, 2003.

[10] Montserrat Batet, David Sánchez, and Aïda Valls. An Ontology-based Measure to Compute Semantic Similarity in Biomedicine. *Biomedical Informatics*, 44(1):118–125, 2011.

[11] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.

[12] Ghassan Beydoun, Graham Low, Numi Tran, and Paul Bogg. Development of a Peer-to-Peer Information Sharing System Using Ontologies. *Expert Systems with Applications*, 38(8):9352–9364, 2011.

[13] Shi Bin, Fang Liying, Yan Jianzhuo, Wang Pu, and Zhao Zhongcheng. Ontology-based Measure of Semantic Similarity between Concepts. In *WRI World Congress on Software Engineering*, volume 2, pages 109–112. IEEE, 2009.

[14] Jürgen Bock and Jan Hettenhausen. MapPSO Results for OAEI 2008. In *The 7th International Semantic Web Conference*, 2008.

[15] Willem Nico Borst. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. Ph.D. Thesis, Institute for Telematica and Information Technology, University of Twente, Netherlands, 1997.

[16] Alexander Budanitsky and Graeme Hirst. Semantic Distance in WordNet: An Experimental, Application-oriented Evaluation of Five Measures. In *The Workshop on WordNet and Other Lexical Resources, Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-2001)*, pages 29–34, 2001.

[17] Caterina Caracciolo, Jérôme Euzenat, Laura Hollink, Ryutaro Ichise, Antoine Isaac, Véronique Malaisé, Christian Meilicke, Juan Pane, Pavel Shvaiko, Heiner Stuckenschmidt, Ondřej Šváb Zamazal, and Vojtěch Svátek. Results of the Ontology Alignment Evaluation Initiative 2008. In *The 7th International Semantic Web Conference*, 2008.

[18] Isabel F. Cruz, Flavio Palandri Antonelli, and Cosmin Stroe. AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. *Proceedings of the VLDB Endowment*, 2(2):1586–1589, 2009.

[19] Jérôme David, Fabrice Guillet, and Henri Briand. Matching Directories and OWL Ontologies with AROMA. In *The 15th ACM International Conference on Information and Knowledge Management*, pages 830–831. ACM, 2006.

[20] Stefan Dessloch, Mauricio A. Hernández, Ryan Wisnesky, Ahmed Radwan, and Jindan Zhou. Orchid: Integrating Schema Mapping and ETL. In *The IEEE 24th International Conference on Data Engineering*, pages 1307–1316. IEEE, 2008.

[21] Robin Dhamankar, Yoonkyong Lee, Anhai Doan, Alon Halevy, and Pedro Domingos. iMAP: Discovering Complex Semantic Matches between Database Schemas. In *The 2004 ACM SIGMOD International Conference on Management of Data*, pages 383–394. ACM, 2004.

[22] Lee R. Dice. Measures of the Amount of Ecologic Association between Species. *Ecology*, 26(3):297–302, 1945.

[23] Hong Hai Do. *Schema Matching and Mapping-based Data Integration*. Ph.D. Thesis, University of Leipzig, Germany, 2005.

[24] Hong Hai Do and E. Rahm. COMA - A System for Flexible Combination of Schema Matching Approaches. In *The 28th International Conference on Very Large Data Bases*, pages 610–621. ACM, 2002.

[25] An Hai Doan and Alon Y. Halevy. Semantic Integration Research in the Database Community: A Brief Survey. *AI Magazine*, 26:83–94, 2005.

[26] An Hai Doan, Jayant Madhavan, Pedro Domingos, and Alon Halevy. Learning to Map between Ontologies on the Semantic Web. In *The 11th International Conference on World Wide Web*, pages 662–673. ACM, 2002.

[27] An Hai Doan, Jayant Madhavan, Pedro Domingos, and Alon Halevy. Ontology Matching: A Machine Learning Approach. In *International Handbooks on Information Systems*, pages 385–403. Springer-Verlag, 2004.

[28] AnHai Doan, Alon Halevy, and Zachary Ives. *Principles of Data Integration*. Morgan Kaufmann, 1st edition, 2012.

[29] Prashant Doshi, Ravikanth Kolli, and Christopher Thomas. Inexact Matching of Ontology Graphs Using Expectation-Maximization. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(2):90–106, 2009.

[30] Marc Ehrig. *Ontology Alignment: Bridging the Semantic Gap*. Springer-Verlag, 2007.

[31] Marc Ehrig and Steffen Staab. QOM - Quick Ontology Mapping. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors, *The Semantic Web - ISWC 2004*, volume 3298 of *LNCS*, pages 683–697. Springer-Verlag, 2004.

[32] Marc Ehrig, Steffen Staab, and York Sure. Bootstrapping Ontology Alignment Methods with APFEL. In *The 4th International Semantic Web Conference*, volume 3729, pages 186–200. Springer-Verlag, 2005.

[33] Marc Ehrig and York Sure. Ontology Mapping - An Integrated Approach. In Christoph J. Bussler, John Davies, Dieter Fensel, and Rudi Studer, editors, *The Semantic Web: Research and Applications*, volume 3053 of *LNCS*, pages 76–91. Springer-Verlag, 2004.

[34] Z. Eidoon, N. Yazdani, and F. Oroumchian. A Vector based Method of Ontology Matching. In *The 3rd International Conference on Semantics, Knowledge and Grid*, pages 378–381. IEEE, 2007.

[35] Jérôme Euzenat. Towards a Principled Approach to Semantic Interoperability. In *Workshop on Ontologies and Information Sharing (IJCAI)*, pages 19–25, 2001.

[36] Jérôme Euzenat, Thanh Le Bach, Jesús Barrasa, Paolo Bouquet, Jan De Bo, Rose Dieng, Marc Ehrig, Manfred Hauswirth, Mustafa Jarrar, Ruben Lara, Diana Maynard, Amedeo Napoli, Giorgos Stamou, Heiner Stuckenschmidt, Pavel Shvaiko, Sergio Tessaris, Sven Van Acker, and Ilya Zaihrayeu. D2.2.3: State of the Art on Ontology Alignment. Technical Report, Knowledge Web Consortium, 2004.

[37] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer-Verlag, 2nd edition, 2013.

[38] Jérôme Euzenat and Petko Valtchev. Similarity-based Ontology Alignment in OWL-Lite. In *European Conference on Artificial Intelligence (ECAI)*, pages 333–337. IOS Press, 2004.

[39] Christiane Fellbaum. WordNet: An Electronic Lexical Database. MIT Press, Cambridge, 1998.

[40] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing Search in Context: The

Concept Revisited. *ACM Transactions on Information Systems*, 20(1):116–131, 2002.

[41] Panorea Gaitanou. Ontologies and Ontology-based Applications. In Miguel-Angel Sicilia and Miltiadis D. Lytras, editors, *Metadata and Semantics*, pages 289–298. Springer-Verlag, 2009.

[42] Fausto Giunchiglia, Fiona McNeill, and Mikalai Yatskevich. Web Service Composition via Semantic Matching of Interaction Specifications. Technical Report, University of Trento, 2006.

[43] Fausto Giunchiglia, Pavel Shvaiko, and Mikalai Yatskevich. Semantic Schema Matching. In Robert Meersman and Zahir Tari, editors, *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*, volume 3760 of *LNCS*, pages 347–365. Springer-Verlag, 2005.

[44] Jorge Gracia and Eduardo Mena. Ontology Matching with CIDER: Evaluation Report for the OAEI 2008. In *The 7th International Semantic Web Conference*, 2008.

[45] Thomas R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[46] Fayçal Hamdi, Haífa Zargayouna, Brigitte Safar, and Chantal Reynaud. TaxoMap in the OAEI 2008 Alignment Contest. In *The 7th International Semantic Web Conference*, 2008.

[47] Adil Hameed, Alun Preece, and Derek Sleeman. Ontology Reconciliation. In *Handbook on Ontologies*, pages 231–250. Springer-Verlag, 2004.

[48] R. W. Hamming. Error Detecting and Error Correcting Codes. *The Bell System Technical Journal*, 29(2):147–160, 1950.

[49] Md. Seddiqui Hanif and Masaki Aono. Metric of Intrinsic Information Content for Measuring Semantic Similarity in an Ontology. In *The 7th Asia-Pacific Conference on Conceptual Modelling (APCCM 2010)*, pages 89–96, 2010.

[50] Babak Bagheri Hariri, Hassan Abolhassani, and Hassan Sayyadi. A Neural-Networks-based Approach for Ontology Alignment. In *The Joint 3rd International Conference on Soft Computing and Intelligent Systems and 7th In-*

ternational Symposium on Advanced Intelligent Systems, pages 1248–1252, 2006.

[51] Michael Hartung, Anika Groß, and Erhard Rahm. COnto–Diff: Generation of Complex Evolution Mappings for Life Science Ontologies. *Biomedical Informatics*, 46(1):15–32, 2013.

[52] Todd Hughes. Introducing I3CON. In *The Information Interpretation and Integration Conference*, 2004.

[53] R Ichise. Machine Learning Approach for Ontology Mapping Using Multiple Concept Similarity Measures. In *The 7th IEEE/ACIS International Conference on Computer and Information Science*, pages 340–346. IEEE, 2008.

[54] Paul Jaccard. The Distribution of the Flora in the Alpine Zone. *The New Phytologist*, 11(2):37–50, 1912.

[55] Prateek Jain, Pascal Hitzler, Amit P. Sheth, Kunal Verma, and Peter Z. Yeh. Ontology Alignment for Linked Open Data. In Peter F. Patel-Schneider, Yue Pan, Pascal Hitzler, Peter Mika, Lei Zhang, Jeff Z. Pan, Ian Horrocks, and Birte Glimm, editors, *The Semantic Web – ISWC 2010*, volume 6496 of *LNCS*, pages 402–417. Springer-Verlag, 2010.

[56] Matthew A. Jaro. Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *The American Statistical Association*, 84(406):414–420, 1989.

[57] Robert Jasper and Mike Uschold. A Framework for Understanding and Classifying Ontology Applications. In *The 12th Workshop on Knowledge Acquisition Modeling and Management*, 1999.

[58] Yves R. Jean-Mary, E. Patrick Shironoshita, and Mansur R. Kabuka. Ontology Matching with Semantic Verification. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):235–251, 2009.

[59] Jay J. Jiang and David W. Conrath. Semantic Similarity based on Corpus Statistics and Lexical Taxonomy. In *The 10th International Conference on Research in Computational Linguistics (ROCLING X)*, pages 19–33, 1997.

[60] David Kensche, Christoph Quix, Mohamed Amine Chatti, and Matthias Jarke. GeRoMe: A Generic Role based Metamodel for Model Management. *Journal on Data Semantics VIII*, pages 82–117, 2007.

[61] David Kensche, Christoph Quix, Xiang Li, and Yong Li. GeRoMeSuite: A System for Holistic Generic Model Management. In *The 33rd International Conference on Very Large Data Bases (VLDB'07)*, pages 1322–1325, 2007.

[62] Toralf Kirsten, Anika Groß, Michael Hartung, and Erhard Rahm. GOMMA: A Component-based Infrastructure for Managing and Analyzing Life Science Ontologies and Their Evolution. *Biomedical Semantics*, 2(6):1–24, 2011.

[63] Josef Kittler, Mohamad Hatef, Robert P.W. Duin, and Jiri Matas. On Combining Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[64] Michel Klein. Combining and Relating Ontologies: An Analysis of Problems and Solutions. In *Workshop on Ontologies and Information Sharing (IJCAI)*, pages 53–62, 2001.

[65] Grzegorz Kondrak. N-Gram Similarity and Distance. In Mariano Consens and Gonzalo Navarro, editors, *String Processing and Information Retrieval*, volume 3772 of *LNCS*, pages 115–126. Springer-Verlag, 2005.

[66] Patrick Lambrix and He Tan. SAMBO - A System for Aligning and Merging Biomedical Ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(3):196–206, 2006.

[67] C. Leacock and M. Chodorow. Combining Local Context and WordNet Similarity for Word Sense Identification. In *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*, pages 265–283, Cambridge, 1998. The MIT Press.

[68] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics Doklady*, 10:707–710, 1966.

[69] Juanzi Li, Tsinghua Univ. Beijing, Jie Tang, Yi Li, and Qiong Luo. RiMOM: A Dynamic Multistrategy Ontology Alignment Framework. *Knowledge and Data Engineering*, 21(8):1218–1232, 2009.

[70] Yuhua Li, Zuhair A. Bandar, and David McLean. An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources. In *IEEE Transactions on Knowledge and Data Engineering*, volume 15, pages 871–882. IEEE, 2003.

[71] Dekang Lin. An Information-Theoretic Definition of Similarity. In *The 15th International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann, 1998.

[72] Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm. Generic Schema Matching with Cupid. In *The 27th International Conference on Very Large Data Bases*, pages 49–58. Morgan Kaufmann, 2001.

[73] Alexander Maedche and Steffen Staab. Measuring Similarity between Ontologies. In Asunción Gómez-Pérez and V. Richard Benjamins, editors, *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, volume 2473 of *LNCS*, pages 251–263. Springer-Verlag, 2002.

[74] Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching. In *The 18th International Conference on Data Engineering*, pages 117–128. IEEE, 2002.

[75] George A. Miller. WordNet: A Lexical Database for English. In *Communications of the ACM*, volume 38, pages 39–41. ACM, 1995.

[76] George A. Miller and Walter G. Charles. Contextual Correlates of Semantic Similarity. In *Language and Cognitive Processes*, volume 6, pages 1–28, 1991.

[77] Prasenjit Mitra and Gio Wiederhold. An Ontology-Composition Algebra. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, pages 93–113. Springer-Verlag, 2004.

[78] Riichiro Mizoguchi. Tutorial on Ontological Engineering Part 2: Ontology Development, Tools and Languages. *New Generation Computing*, 22(1):61–96, 2004.

[79] Miklos Nagy, Maria Vargas-Vera, and Piotr Stolarski. DSSim Results for OAEI 2008. In *The 7th International Semantic Web Conference*, 2008.

[80] Saul B. Needleman and Christian D. Wunsch. A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. *Molecular Biology*, 48:443–453, 1970.

[81] Azadeh Haratian Nezhadi, Bita Shadgar, and Alireza Osareh. Ontology Alignment Using Machine Learning Techniques. *International Journal of Computer Science and Information Technology*, 3(2):139–150, 2011.

[82] Thi Thuy Anh Nguyen and Stefan Conrad. A New Structure-based Similarity Measure for Automatic Ontology Matching. In *The 4th International Conference on Knowledge Discovery and Information Retrieval*, pages 443–449. SciTePress, 2012.

[83] Thi Thuy Anh Nguyen and Stefan Conrad. Combination of Lexical and Structure-based Similarity Measures to Match Ontologies Automatically. In Ana Fred, Jan L. G. Dietz, Kecheng Liu, and Joaquim Filipe, editors, *Knowledge Discovery, Knowledge Engineering and Knowledge Management*, volume 415 of *LNCS*, pages 101–112. Springer-Verlag, 2013.

[84] Thi Thuy Anh Nguyen and Stefan Conrad. A Semantic Similarity Measure between Nouns based on the Structure of WordNet. In *The 15th International Conference on Information Integration and Web-based Applications & Services (iiWAS2013)*, pages 605–609. ACM, 2013.

[85] Thi Thuy Anh Nguyen and Stefan Conrad. Applying Information-Theoretic and Edit Distance Approaches to Flexibly Measure Lexical Similarity. In *The 6th International Conference on Knowledge Discovery and Information Retrieval*, pages 505–511. SciTePress, 2014.

[86] Thi Thuy Anh Nguyen and Stefan Conrad. An Improved String Similarity Measure based on Combining Information-Theoretic and Edit Distance Methods. In Ana Fred, Jan L. G. Dietz, David Aveiro, Kecheng Liu, and Joaquim Filipe, editors, *Knowledge Discovery, Knowledge Engineering and Knowledge Management*, volume 553 of *LNCS*, pages 228–239. Springer-Verlag, 2015.

[87] Thi Thuy Anh Nguyen and Stefan Conrad. Ontology Matching Using Multiple Similarity Measures. In *The 7th International Conference on Knowledge Discovery and Information Retrieval*, pages 603–611. SciTePress, 2015.

[88] Natalya Fridman Noy and Michel Klein. Ontology Evolution: Not the Same as Schema Evolution. *Knowledge and Information Systems*, 6(4):428–440, 2004.

[89] Natalya Fridman Noy and Mark A. Musen. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In *The National Conference on Artificial Intelligence (AAAI)*, pages 450–455, 2000.

[90] Natalya Fridman Noy and Mark A. Musen. Anchor-PROMPT: Using Non-Local Context for Semantic Matching. In *Workshop on Ontologies and Information Sharing at the 17th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 63–70, 2001.

[91] Natalya Fridman Noy and Mark A. Musen. Evaluating Ontology-Mapping Tools: Requirements and Experience. In *Ontoweb-Sig3 Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management*, pages 1–14, 2002.

[92] Natalya Fridman Noy and Mark A. Musen. PROMPTDIFF: A Fixed-Point Algorithm for Comparing Ontology Versions. In *The 8th National Conference on Artificial Intelligence*, pages 744–750, 2002.

[93] Natalya Fridman Noy and Mark A. Musen. Ontology Versioning in an Ontology Management Framework. *Intelligent Systems*, 19(4):6–13, 2004.

[94] Rahul Parundekar, Craig A. Knoblock, and José Luis Ambite. Linking and Building Ontologies of Linked Data. In Peter F. Patel-Schneider, Yue Pan, Pascal Hitzler, Peter Mika, Lei Zhang, Jeff Z. Pan, Ian Horrocks, and Birte Glimm, editors, *The Semantic Web – ISWC 2010*, volume 6496 of *LNCS*, pages 598–614. Springer-Verlag, 2010.

[95] Rahul Parundekar, Craig A. Knoblock, and José Luis Ambite. Discovering Concept Coverings in Ontologies of Linked Data Sources. In Philippe et al. Cudré-Mauroux, editor, *The Semantic Web – ISWC 2012*, volume 7649 of *LNCS*, pages 427–443. Springer-Verlag, 2012.

[96] Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. Using Measures of Semantic Relatedness for Word Sense Disambiguation. In *The 4th International Conference on Computational Linguistics and Intelligent Text Processing*, volume 2588, pages 241–257. Springer-Verlag, 2003.

[97] Siddharth Patwardhan and Ted Pedersen. Using WordNet-based Context Vectors to Estimate the Semantic Relatedness of Concepts. In *The EACL Workshop on Making Sense of Sense: Bringing Psycholinguistics and Computational Linguistics Together*, pages 1–8. IEEE, 2006.

[98] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. WordNet:: Similarity - Measuring the Relatedness of Concepts. In *19th National Conference on Artificial Intelligence (AAAI-04)*, pages 1024–1025, Cambridge, 2004. AAAI.

[99] Giuseppe Pirró and Jérôme Euzenat. A Feature and Information Theoretic Framework for Semantic Similarity and Relatedness. In *The 9th International Semantic Web Conference on The Semantic Web*, pages 615–630. Springer-Verlag, 2010.

[100] Giuseppe Pirró and Nuno Seco. Design, Implementation and Evaluation of a New Semantic Similarity Metric Combining Features and Intrinsic Information Content. In Robert Meersman and Zahir Tari, editors, *On the Move to Meaningful Internet Systems: OTM 2008*, volume 5332 of *LNCS*, pages 1271–1288. Springer-Verlag, 2008.

[101] Livia Predoiu, Cristina Feier, Francois Scharffe, Jos de Bruijn, Francisco Martín-Recuerda, Dimitar Manov, and Marc Ehrig. D4.2.2: State-of-the-Art Survey on Ontology Merging and Aligning V2. Technical Report, SEKT Consortium, 2005.

[102] Livia Predoiu, Francisco Martín-Recuerda, Axel Polleres, Cristina Feier, Adrian Mocan, Jos de Bruijn, Fabio Porto, Doug Foxvog, and Kerstin Zimmermann. D1.5: Framework for Representing Ontology Networks with Mappings that Deal with Conflicting and Complementary Concept Definitions. Technical Report, DERI, 2004.

[103] Roy Rada, Hafedh Mili, Ellen Bicknell, and Maria Blettner. Development and Application of a Metric on Semantic Nets. *IEEE Transactions on Systems, Man and Cybernetics*, 19(1):17–30, 1989.

[104] Erhard Rahm and Philip A. Bernstein. A Survey of Approaches to Automatic Schema Matching. *The International Journal on Very Large Data Bases*, 10(4):334–350, 2001.

[105] Philip Resnik. Semantic Similarity in a Taxonomy: An Information-based Measure and its Application to Problems of Ambiguity in Natural Language. *Artificial Intelligence Research*, 11:95–130, 1999.

[106] Ray Richardson and Alan F. Smeaton. Using WordNet in a Knowledge-based Approach to Information Retrieval. In *Working Paper, CA-0395*, Ireland, 1995. Dublin City University, School of Computer Applications.

[107] C. J. van Rijsbergen. *Information Retrieval*. London: Butterworths, 2nd edition, 1979.

[108] Dave Robertson, Fausto Giunchiglia, Frank van Harmelen, Maurizio Marchese, Marta Sabou, Marco Schorlemmer, Nigel Shadbolt, Ronnie Siebes, Carles Sierra, Chris Walton, Srinandan Dasmahapatra, Dave Dupplaw, Paul Lewis, Mikalai Yatskevich, Spyros Kotoulas, Adrian Perreau de Pinninck, and Antonis Loizou. Open Knowledge Semantic Webs Through Peer-to-Peer Interaction. Technical Report, University of Trento, 2006.

[109] John F. Roddick. A Survey of Schema Versioning Issues for Database Systems. *Information and Software Technology*, 37:383–393, 1995.

[110] M. Andrea Rodríguez and Max J. Egenhofer. Determining Semantic Similarity among Entity Classes from Different Ontologies. In *IEEE Transactions on Knowledge and Data Engineering*, volume 15, pages 442–456. IEEE, 2003.

[111] Herbert Rubenstein and John B. Goodenough. Contextual Correlates of Synonymy. In *Communications of the ACM*, volume 8, pages 627–633. ACM, 1965.

[112] Marta Sabou and Jorge Gracia. Spider: Bringing Non-Equivalence Mappings to OAEI. In *The 7th International Semantic Web Conference*, 2008.

[113] David Sánchez and Montserrat Batet. Semantic Similarity Estimation in the Biomedical Domain: An Ontology-based Information-Theoretic Perspective. *Biomedical Informatics*, 44(5):749–759, 2011.

[114] David Sánchez, Montserrat Batet, David Isern, and Aïda Valls. Ontology-based Semantic Similarity: A New Feature-based Approach. *Expert Systems with Applications*, 39(9):7718–7728, 2012.

[115] Nuno Seco, Tony Veale, and Jer Hayes. An Intrinsic Information Content Metric for Semantic Similarity in WordNet. In *The 16th Eureopean Conference on Artificial Intelligence*, pages 1089–1090, 2004.

[116] Md. Hanif Seddiqui and Masaki Aono. An Efficient and Scalable Algorithm for Segmented Alignment of Ontologies of Arbitrary Size. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(4):344–356, 2009.

[117] Len Seligman, Peter Mork, Alon Halevy, Ken Smith, Michael J. Carey, Kuang Chen, Chris Wolf, Jayant Madhavan, and Akshay Kannan. OpenII: An Open Source Information Integration Toolkit. In *The 2010 ACM SIGMOD International Conference on Management of Data*, pages 1057–1060. ACM, 2010.

[118] A. Sharma. Ontology Matching Using Weighted Graphs. In *The 1st International Conference on Digital Information Management*, pages 121–124. IEEE, 2006.

[119] Pavel Shvaiko and Jérôme Euzenat. A Survey of Schema-based Matching Approaches. *Data Semantics*, 4:146–171, 2005.

[120] Thabet Slimani, Ben Yaghlane Boutheina, and Khaled Mellouli. A New Similarity Measure based on Edge Counting. In *World Academy of Science, Engineering and Technology*, pages 34–38, 2006.

[121] Barry Smith and Christopher Welty. Ontology: Towards a New Synthesis. In *The International Conference on Formal Ontology in Information Systems*, pages 3–9. ACM, 2001.

[122] Steffen Staab and Heiner Stuckenschmidt. *Semantic Web and Peer-to-Peer*. Springer-Verlag, 2006.

[123] Steffen Staab and Rudi Studer. *Handbook on Ontologies*. Springer-Verlag, 2nd edition, 2009.

[124] Umberto Straccia and Raphaël Troncy. oMAP: Combining Classifiers for Aligning Automatically OWL Ontologies. In *The 6th International Conference on Web Information Systems Engineering*, volume 3806, pages 133–147. Springer-Verlag, 2005.

[125] Rudi Studer, V. Richard Benjamins, and Dieter Fensel. Knowledge Engineering: Principles and Methods. *Data & Knowledge Engineering*, 25(1-2):161–197, 1998.

[126] William Sunna and Isabel F. Cruz. Structure-based Methods to Enhance Geospatial Ontology Alignment. In *The 2nd International Conference on GeoSpatial Semantics*, pages 82–97. Springer-Verlag, 2007.

[127] Michael Sussna. Word Sense Disambiguation for Free-Text Indexing Using a Massive Semantic Network. In *The 2nd International Conference on Information and Knowledge Management*, pages 67–74. ACM, 1993.

[128] Mohamed Ali Hadj Taieb, Mohamed Ben Aouicha, Mohamed Tmar, and Abdelmajid Ben Hamadou. Wikipedia Category Graph and New Intrinsic Information Content Metric for Word Semantic Relatedness Measuring. In Yang Xiang, Mukaddim Pathan, Xiaohui Tao, and Hua Wang, editors, *Data and Knowledge Engineering*, volume 7696 of *LNCS*, pages 128–140. Springer-Verlag, 2012.

[129] Jie Tang, Juanzi Li, Bangyong Liang, Xiaotong Huang, Yi Li, and Kehong Wang. Using Bayesian Decision for Ontology Mapping. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(4):243–262, 2006.

[130] Konstantin Todorov. *Ontology Matching by Combining Instance-based Concept Similarity Measures with Structure*. Ph.D. Thesis, University of Osnabrück, Germany, 2009.

[131] Hong-Minh Tran and Dan Smith. Word Similarity in WordNet. In *The 3rd International Conference on High Performance Scientific Computing*, pages 293–302. Springer-Verlag, 2008.

[132] Kagan Tumer and Joydeep Ghosh. Classifier Combining: Analytical Results and Implications. In *The AAAI-96 Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms*, pages 126–132. AAAI Press, 1995.

[133] Amos Tversky. Features of Similarity. *Psychological Review*, 84(4):327–352, 1977.

[134] Pepijn R.S. Visser, Dean M. Jones, Trevor J.M. Bench-Capon, and M.J.R. Shave. An Analysis of Ontology Mismatches: Heterogeneity versus Interoperability. In *AAAI 1997 Spring Symposium on Ontological Engineering*, 1997.

[135] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner. Ontology-based Integration of Information - A Survey of Existing Approaches. In *Workshop on Ontologies and Information Sharing (IJCAI)*, pages 108–117, 2001.

[136] Shen Wan and Rafal A. Angryk. Measuring Semantic Similarity Using WordNet-based Context Vectors. In *The IEEE International Conference on Systems, Man and Cybernetics*, pages 908–913. IEEE, 2007.

[137] Peng Wang and Baowen Xu. Lily: Ontology Alignment Results for OAEI 2008. In *The 7th International Semantic Web Conference*, 2008.

[138] Xia Wang, Yihong Ding, and Yi Zhao. Similarity Measurement about Ontology-based Semantic Web Services. In *The Workshop on Semantics for Web Services*, 2006.

[139] Ying Wang, Weiru Liu, and David A. Bell. A Structure-based Similarity Spreading Approach for Ontology Matching. In *The 4th International Conference on Scalable Uncertainty Management*, pages 361–374. Springer-Verlag, 2010.

[140] Floris Wiesman, Nico Roos, and Paul Vogt. Automatic Ontology Mapping for Agent Communication. In *The First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 563–564. ACM, 2002.

[141] William E. Winkler. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In *The Section on Survey Research*, pages 354–359, 1990.

[142] Zhibiao Wu and Martha Palmer. Verbs Semantics and Lexical Selection. In *The 32nd Annual Meeting on Association for Computational Linguistics*, volume 6 of *Association for Computational Linguistics*, pages 133–138, 1994.

[143] Dongqiang Yang and David M. W. Powers. Measuring Semantic Similarity in the Taxonomy of WordNet. In *The 28th Australasian Conference on Computer Science ACSC*, volume 38, pages 315–322, Australia, 2005. Australian Computer Society.

[144] Katrin Zaiß. *Instance-based Ontology Matching and the Evaluation of Matching Systems*. Ph.D. Thesis, Heinrich-Heine-Universität Düsseldorf, Germany, 2011.

[145] Katrin Zaiß and Stefan Conrad. Partial Ontology Matching Using Instance Features. In Robert Meersman, Tharam Dillon, and Pilar Herrero, editors, *On the Move to Meaningful Internet Systems: OTM 2009*, volume 5871 of *LNCS*, pages 1201–1208. Springer-Verlag, 2009.

[146] Katrin Zaiß, Stefan Conrad, and Sven Vater. A Benchmark for Testing Instance-based Ontology Matching Methods. In *The 17th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2010)*. Posters and Demos, 2010.

[147] Haïfa Zargayouna, Brigitte Safar, and Chantal Reynaud. TaxoMap in the OAEI 2007 Alignment Contest. In *The 6th International Semantic Web Conference*, 2007.