Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros de Telecomunicación

# SEMANTIC SIMILARITY ANALYSIS AND APPLICATION IN KNOWLEDGE GRAPHS

Tesis Doctoral

Ganggao Zhu
Máster en Software y Sistemas

2017

Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros de Telecomunicación

# SEMANTIC SIMILARITY ANALYSIS AND APPLICATION IN KNOWLEDGE GRAPHS

Tesis Doctoral

Ganggao Zhu
Máster en Software y Sistemas

2017

Departamento de Ingeniería de Sistemas Telemáticos

Escuela Técnica Superior de Ingenieros de Telecomunicación

Universidad Politécnica de Madrid

# SEMANTIC SIMILARITY ANALYSIS AND APPLICATION IN KNOWLEDGE GRAPHS

Autor:

Ganggao Zhu
Máster en Software y Sistemas

Tutor:

Carlos Ángel Iglesias Fernández
Doctor Ingeniero de Telecomunicación

2017

Tribunal nombrado por el Magfco. y Excmo. Sr. Rector de la Universidad Politécnica de Madrid, el día ________ de ________________ de __________.

**Presidente:** ________________________________________________

**Vocal:** ________________________________________________

**Vocal:** ________________________________________________

**Vocal:** ________________________________________________

**Secretario:** ________________________________________________

**Suplente:** ________________________________________________

**Suplente:** ________________________________________________

Realizado el acto de defensa y lectura de la Tesis el día ________ de ________________ de __________. en la E.T.S.I. Telecomunicación habiendo obtenido la calificación de ____________________________________________________________________.

EL PRESIDENTE                                        LOS VOCALES

EL SECRETARIO

A mi madre, a mi padre, gracias por todo.

# Acknowledgements

Foremost, I would like to express my sincere gratitude to my adviser, Carlos A. Iglesias, for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. He provided me an excellent freedom atmosphere for doing research and implementing new ideas. His excellent guidance helped me in all the time of research and writing of this thesis.

Besides, I would like to thank my fellow labmates for the discussions, lunch talk, pizza time, morning coffee game, working together before deadlines, and for all the fun we have had in the last five years. It was fantastic to have the opportunity to work with Álvaro, Miguel, Fernando, Óscar, Adam, Emilio, Geovanny, and Shengjing.

Also, a very special gratitude goes out to all my good friends in Madrid, who were willing to help and give their best suggestions. It would have been a lonely time without Zhi, Bo, Xian, Jiayao, Rong, Jianguo, Qiong, Meijuan, Shan, Jie, Qi, Yu, Pengming, Sisi, Yongjun, Baojun, Irene, Angel and many others. It was great, fantastic, wonderful six years, having all the stimulating discussions, sleepless party nights, festival celebrations, Chinese food, games and all the fun.

Last but not the least, I would like to thank my parents. They were always supporting me and encouraging me with their best wishes. Without their precious support it would not be possible for me to study abroad and to conduct this research. I am also grateful to my other family members and friends in China who have encouraged and supported me along the way.

My deepest appreciation for all your encouragement and help!

# Resumen

Las técnicas avanzadas de extracción de información y la creciente disponibilidad de datos vinculados han dado a luz a la noción de Grafo de Conocimiento (Knowledge Graph, KG) de gran escala. Con la creciente popularidad de KGs que contienen millones de conceptos y entidades, la investigación de herramientas fundamentales que estudian características semánticas de KGs es crítica para el desarrollo de aplicaciones basadas en KG, aparte del estudio de las técnicas de población de KG. Con este enfoque, esta tesis explora la similitud semántica en KGs teniendo en cuenta el concepto de taxonomía, concepto de distribución, la entidad descripciones y las categorías.

La similitud semántica captura la cercanía de significados. A través del estudio de la red semántica de conceptos y entidades con relaciones significativas en KGs, hemos propuesto una nueva métrica de semántica WPath semántica, y un nuevo método de computación basado en información gráfica (IC). Con el WPath y el IC basado en gráfos, la similitud semántica de los conceptos se puede calcular directamente, basándose únicamente en el conocimiento estructural y el conocimiento estadístico contenido en KGs. Los experimentos en similitud de palabras han demostrado que la mejora de los métodos propuestos es estadísticamente significativa en comparación con los métodos convencionales. Por otra parte, observando que los conceptos suelen ser colocados con descripciones textuales, proponemos un nuevo enfoque de incorporación para formar el concepto y incorporación de palabras conjuntamente. El espacio vectorial compartido de conceptos y palabras ha proporcionado una computación de la similitud conveniente entre conceptos y palabras a través de similitud vectorial. De manera adicional, se ilustran algunas aplicaciones de modelos basados en el conocimiento, en corpus y en embeddings en la tarea de desambiguación y clasificación semántica, con el fin de demostrar la capacidad e idoneidad de diferentes métodos de similitud en aplicaciones específicas. Por último, la búsqueda de entidad semántica se utiliza como una demostración ilustrativa de un nivel más alto de la aplicación que consiste en similitud basado en el texto de concordancia, la desambiguación y la expansión de la consulta. Para implementar la demostración completa de la consulta de información centrada en la entidad, también proponemos un enfoque basado en reglas para construir y ejecutar automáticamente consultas SPARQL.

# Abstract

The advanced information extraction techniques and increasing availability of linked data have given birth to the notion of large scale Knowledge Graph (KG). With the increasing popularity of KGs containing millions of concepts and entities, the research of fundamental tools studying semantic features of KGs is critical for the development of KG-based applications, apart from the study of KG population techniques. With such focus, this thesis exploits semantic similarity in KGs taking into consideration of concept taxonomy, concept distribution, entity descriptions and categories.

Semantic similarity captures the closeness of meanings. Through studying the semantic network of concepts and entities with meaningful relations in KGs, we proposed a novel WPath semantic similarity metric and new graph-based Information Content (IC) computation method. With the WPath and graph-based IC, semantic similarity of concepts can be computed directly and only based on the structural and statistical knowledge contained in KG. The word similarity experiments have shown that the improvement of the proposed methods is statistical significant comparing to conventional methods. Moreover, observing that concepts are usually collocated with textual descriptions, we propose a novel embedding approach to train concept and word embedding jointly. The shared vector space of concepts and words, has provided convenient similarity computation between concepts and words through vector similarity. Furthermore, the applications of knowledge-based, corpus-based and embedding-based similarity methods are shown and compared in the task of semantic disambiguation and classification, in order to demonstrate the capability and suitability of different similarity methods in specific application. Finally, semantic entity search is used as an illustrative showcase to demonstrate higher level of the application consisting of text matching, disambiguation and query expansion. To implement the complete demonstration of entity-centric information querying, we also propose a rule-based approach for constructing and executing SPARQL queries automatically.

In summary, the thesis exploits various similarity methods and illustrates their corresponding applications for KGs. The proposed similarity methods and presented similarity-based applications would help in facilitating the research and development of applications in KGs.

# Contents

CHAPTER 1

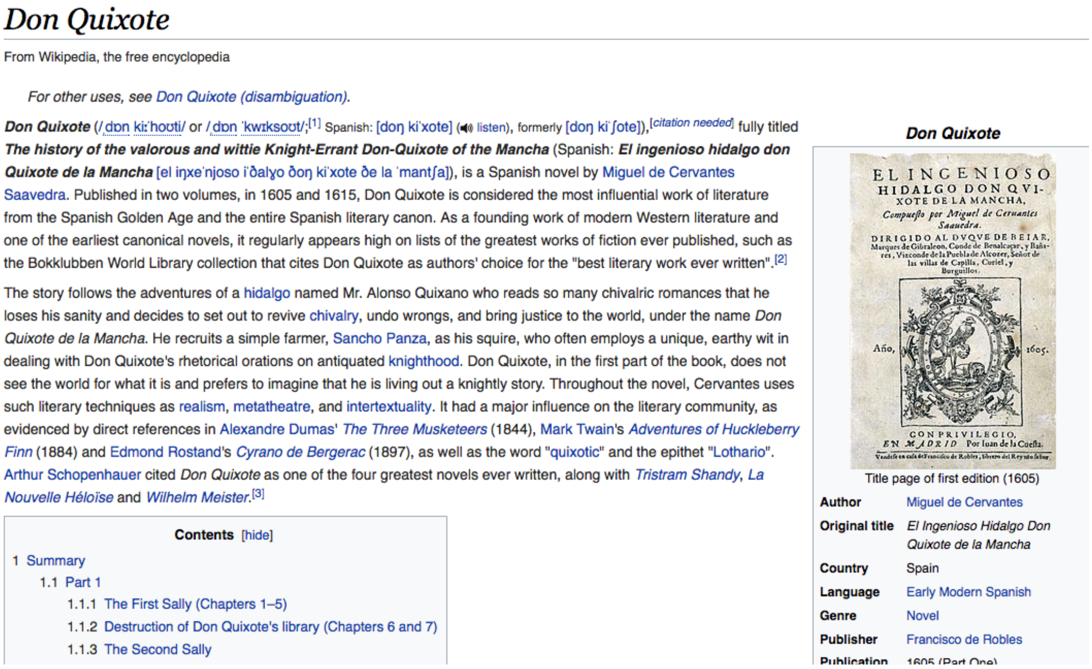
# Introduction

*The recent advances in knowledge representation and organization have given birth to large scale KGs containing millions of concepts, entities and their relationships. Semantic similarity is an important metric to quantify how much those concepts and entities are alike to each other respect to their meanings. This thesis investigates semantic similarity metrics leveraging the semantic information contained in KG. Moreover, we also study how to use semantic similarity to develop applications such as disambiguation, classification and search for concepts and entities in KG.*

*In this chapter, the thesis motivation, objectives and solution architecture are presented to the readers. We summarize the research problems including: (1) semantic similarity of concepts, words and entities; (2) word and named entity disambiguation; (3) ontological concept classification and concept-based entity search. Finally, we set up objectives of answering specific research problems, and outline the solutions proposed to achieve those objectives.*

## 1.1 Motivation

The increasing availability of Linked Open Data (LOD) (Bizer et al., 2009a) has given birth to the notion of modern large scale KGs which contain millions of entities and their relationships, with popular examples such as Freebase (Bollacker et al., 2008), DBpedia (Bizer et al., 2009b), and YAGO (Hoffart et al., 2013). Such KGs have transformed the web from a web of documents into a web of entities applying the advanced information extraction techniques (Banko et al., 2007) and Semantic Web (Berners-Lee et al., 2001) techniques. With such transformation, the fine-grained graph representation of entity knowledge has improved the connectivity and accessibility of entity-centric information.



*Don Quixote*

From Wikipedia, the free encyclopedia

*For other uses, see Don Quixote (disambiguation).*

***Don Quixote*** (/ˌdɒn kiːˈhoʊti/ or /ˌdɒn ˈkwɪksoʊt/;[1] Spanish: [doŋ kiˈxote] (listen), formerly [doŋ kiˈʃote]),[citation needed] fully titled ***The history of the valorous and wittie Knight-Errant Don-Quixote of the Mancha*** (Spanish: ***El ingenioso hidalgo don Quixote de la Mancha*** [el iŋxeˈnjoso iˈðalɣo ðoŋ kiˈxote ðe la ˈmantʃa]), is a Spanish novel by Miguel de Cervantes Saavedra. Published in two volumes, in 1605 and 1615, Don Quixote is considered the most influential work of literature from the Spanish Golden Age and the entire Spanish literary canon. As a founding work of modern Western literature and one of the earliest canonical novels, it regularly appears high on lists of the greatest works of fiction ever published, such as the Bokklubben World Library collection that cites Don Quixote as authors' choice for the "best literary work ever written".[2]

The story follows the adventures of a hidalgo named Mr. Alonso Quixano who reads so many chivalric romances that he loses his sanity and decides to set out to revive chivalry, undo wrongs, and bring justice to the world, under the name *Don Quixote de la Mancha*. He recruits a simple farmer, Sancho Panza, as his squire, who often employs a unique, earthy wit in dealing with Don Quixote's rhetorical orations on antiquated knighthood. Don Quixote, in the first part of the book, does not see the world for what it is and prefers to imagine that he is living out a knightly story. Throughout the novel, Cervantes uses such literary techniques as realism, metatheatre, and intertextuality. It had a major influence on the literary community, as evidenced by direct references in Alexandre Dumas' *The Three Musketeers* (1844), Mark Twain's *Adventures of Huckleberry Finn* (1884) and Edmond Rostand's *Cyrano de Bergerac* (1897), as well as the word "quixotic" and the epithet "Lothario". Arthur Schopenhauer cited *Don Quixote* as one of the four greatest novels ever written, along with *Tristram Shandy*, *La Nouvelle Héloïse* and *Wilhelm Meister*.[3]

**Contents** [hide]

1 Summary
  1.1 Part 1
    1.1.1 The First Sally (Chapters 1–5)
    1.1.2 Destruction of Don Quixote's library (Chapters 6 and 7)
    1.1.3 The Second Sally

***Don Quixote***

EL INGENIOSO
HIDALGO DON QVIXOTE DE LA MANCHA,
Compueſto por Miguel de Ceruantes Saauedra.
DIRIGIDO AL DVQVE DE BEIAR,
Marques de Gibraleon, Conde de Benalcaçar, y Bañares, Vizconde de la Puebla de Alcozer, Señor de las villas de Capilla, Curiel, y Burguillos.
Año, 1605.
CON PRIVILEGIO,
EN MADRID Por Iuan de la Cueſta.
Vendeſe en caſa de Franciſco de Robles, librero del Rey nro ſeñor

Title page of first edition (1605)

| | |
|---|---|
| **Author** | Miguel de Cervantes |
| **Original title** | *El Ingenioso Hidalgo Don Quixote de la Mancha* |
| **Country** | Spain |
| **Language** | Early Modern Spanish |
| **Genre** | Novel |
| **Publisher** | Francisco de Robles |
| **Publication** | 1605 (Part One) |

Figure 1.1: Wikipedia Article about Don Quixote

For example, Figure 1.1 shows the Wikipedia document about Don Quixote in unstructured textual form, while Figure 1.2 shows DBpedia entity *dbr:Don_ Quixote*[1] which can be connected with other entities through various semantic relationships. Moreover, the entities in green nodes are described with meaningful concepts in blue nodes via a special relation rdf:type. Those concepts representing conceptual abstractions of things (e.g. dbo:Book) group different entities sharing similar characteristics together with well defined concept taxonomy, thus, entities can also be retrieved from KG through meaningful concepts.

[1] dbr is an abbreviation which is called namespace prefix. We abbreviate Uniform Resource Identifier (URI) namespaces with common prefixes, see http://prefix.cc for details.

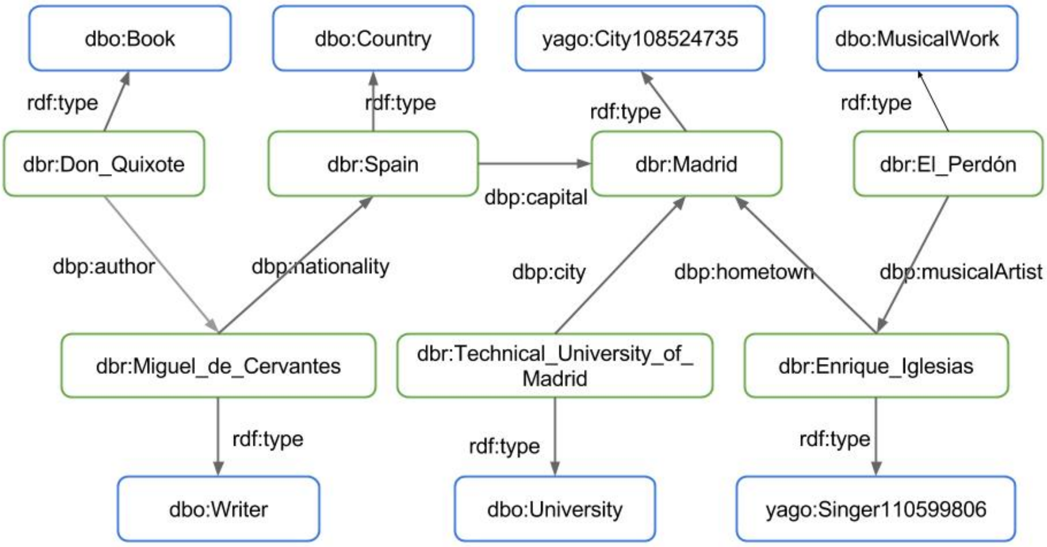


Figure 1.2: A small subgraph of DBpedia related to Don Quixote, Madrid and Spain

All these improvements of information management in KGs, have provided novel opportunities to facilitate many different Natural Language Processing (NLP) and Information Retrieval (IR) tasks (Hovy et al., 2013) including text analysis (Meij et al., 2012), document retrieval (Medelyan et al., 2008), entity linking (Shen et al., 2015), Word Sense Disambiguation (WSD) (Navigli, 2009; Moro et al., 2014), Named Entity Disambiguation (NED) (Hoffart et al., 2012a; Hulpus et al., 2015), query interpretation (Pound et al., 2010a), document modeling (Schuhmacher and Ponzetto, 2014) and Question Answering (QA) (Shekarpour et al., 2015) to name a few. Furthermore, as the increasing amount of structured data has become available in KGs, the advanced KG-based applications are emerging and gearing toward entity-centric applications. However, several research problems are common to make the development of KG-based applications difficult.

**(1) How to compute semantic similarity of concepts in KG?** Measuring semantic similarity between concepts in the lexical database WordNet (Miller, 1995) is an important task because concept similarity is a foundation of computing word similarity and sentence similarity, as well as analyzing textual document (Mihalcea and Tarau, 2004). This pipeline for concept similarity processing can be used to concepts in KG. As shown in Figure 1.3, concept similarity can be used to compute entity similarity, while the hierarchical relations between concepts encoded in semantic similarity are useful for applications such as concept expansion and concept-based retrieval (Dragoni et al., 2012). In general, semantic similarity metrics can be used for weighting or ranking similar concepts based on a concept taxonomy. In such way, semantic similarity methods could be applied in KGs for concept-based entity retrieval or QA, where those entities that contain types having similar meaning to query

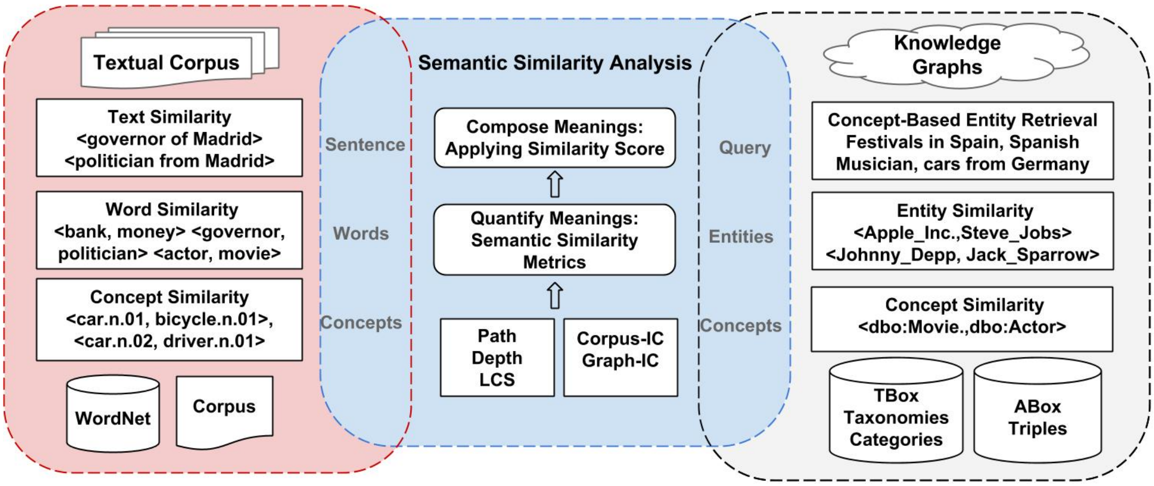


Figure 1.3: The Motivation of Applying Semantic Similarity Analysis to Knowledge Graphs.

concepts would be retrieved. Furthermore, in entity modeling, semantic similarity could be used to cluster entities based on their concepts. The conventional semantic similarity methods and tools are designed and implemented for a specific taxonomy such as WordNet. Those methods and tools cannot be directly applied to various KGs having different domain ontologies and concept taxonomies. Thus the adaptation of the conventional concept similarity method to modern KGs becomes important. Especially, recent efforts have transformed WordNet to be accessed and applied as a concept taxonomy in KGs by converting the conventional representation of WordNet into a novel linked data representation. For example, KGs such as DBpedia, YAGO and BabelNet (Navigli and Ponzetto, 2012) have integrated WordNet and used it as part of their concept taxonomy to categorize entity instances into different types. In consequence, the adaptation of conventional semantic similarity methods to compute concept similarity in KGs would be beneficial to a wide range of applications.

**(2) How to discriminate words and named entities with KG?** A key challenge for processing natural language texts based on KG is the ambiguity of words and entity names. For example, the polysemous word bank can refer to multiple meanings such as a *repository for money* or a *pile of earth on the edge of a river*, while a name "Michael Jordan" can link to multiple entities registered in DBpedia, such as the famous basketball player *dbr:Michael_ Jordan* or professor *dbr:Michael_ I._ Jordan*, which are illustrated in Table 1.1. Since polysemous words and entity names have multiple entries in KG, mapping them from text to corresponding concepts or entities in KG need to perform the task of disambiguation with the KG. These two tasks are commonly called Word Sense Disambiguation and Named Entity Disambiguation respectively. Both tasks need to select the correct entry of concept and entities in KG according to the concept or entity mention context. Studying the sim-

| | |
|---|---|
| Entity | dbr:Michael_Jordan |
| Abstract | Michael Jeffrey Jordan (born February 17, 1963), also known by his initials, MJ, is an American former professional basketball player......Jordan became the first billionaire NBA player in history. |
| Entity | dbr:Michael_I._Jordan |
| Abstract | Michael Irwin Jordan (born 1956) is an American scientist, Professor at the University of California, Berkeley and leading researcher in machine learning and artificial intelligence. |

Table 1.1: The Examples of Named Entities Matching "Michael Jordan" in DBpedia.

ilarity between the mention's surrounding context with an ambiguous concept or entity is the key problem to design a robust disambiguation method. Thus, following the similarity study in KG, we exploit the research problem of using similarity-based disambiguation methods for discriminating words and named entities, in order to facilitate those KG-based applications requiring WSD or NED.

**(3) How to use similarity as a feature for classification?** Classifying a text into a proper predefined ontology class is a common problem in text analysis and KG applications. For example, in case of concept-level sentiment analysis for restaurant reviews, the application needs to predict the polarity (e.g. positive or negative) of a restaurant entity in terms of its food or drink aspects. In order to perform the correct sentiment analysis, a concept classifier is needed to classify those words or short texts such as pasta, noodle, steak, tea, and wine into their ontological parent concepts, FOOD and DRINK. In case of entity type recognition, given the abstract of entity *dbr:Michael_I._ Jordan*, a concept classifier is needed to give the entity type that is defined in a ontology such as *dbo:Scientist* according to specific application and the background KG. Many applications need to classify unseen entities or textual mentions into proper ontological classes, while the current existing common approaches to implement such concept classification system usually rely on Bag of Words (BOW) representation which is not able to address vocabulary mismatch, and difficult to be applied to new domains. Furthermore, most of the classification features can not represent hierarchical relations between concepts when concepts are required to be classified into their ontological parents. We observe that semantic similarity provides a semantic feature set that cannot only give correspondence weight between input words with feature words, but also encodes the hierarchical relation between concepts. As a consequence, the research problem of using similarity as feature for classification needs to be investigated.

**(4) How to answer entity-centric natural language keyword queries from KG?** Answering more complex queries for entity-centric information can be difficult and even impossible with simple textual retrieval. For example, retrieving a list of lakes or mountains in Spain is beyond the capability of existing Web query engines built on top of BOW models, despite the fact that the relevant information is available in Web of documents. As modern KGs represent entities in Resource Description Framework (RDF), the RDF triple query language SPARQL can be used to match information conveniently in KGs. However, SPARQL queries are needed to specify the correct semantic resources in the correct position of triple patterns in form of *subject, predicate* and *object*, which is tedious for users to locate those semantic resources in large scale KGs. Natural language interfaces are much more natural for querying information from KGs, whereas, automatic semantic parsing and translating of natural language queries into SPARQL queries is a difficult task. A semantic parsing system would facilitate many application seeking to provide natural language interfaces for KGs. Therefore, the research problem of semantic parsing of natural language queries to SPARQL queries is studied to offer easy tools for developing natural language interface for querying various KGs.

In conclusion, due to the increasing popularity, growth and convenience of using KG techniques for managing information, developing fundamental tools for KG based applications is an interesting research field. Accordingly, our motivation of the following thesis was to propose and implement solutions of those aforementioned research problems in developing new knowledge-based applications taking advantage of modern KGs. In particular, we are motivated by the fact that there is still a lack of useful similarity tools and metrics, as well as corresponding similarity-based applications to solve the research problems of ambiguity, classification and semantic entity search.

## 1.2 Objectives

The primary objective of the thesis is to deliver useful tools that can address fundamental problems and facilitate the development of applications with KGs. By referring to the previous section, the issues in applying KGs originate from a number of causes in different application layer. Therefore, we have decomposed the thesis global objective into a number of more specific ones in order to build the final solution step by step:

- **(1) Develop a semantic similarity framework for KGs**.
  Our objective is to define a semantic similarity method that can be used to measure semantic similarity between concepts in KGs only based on the semantic knowledge contained in KGs itself. Thus, the semantic similarity method can be used in different KGs and applications. Apart from the applicability of semantic similarity method, it should have comparatively better performance than other existing methods.

- **(2) Develop a disambiguation framework for KGs**.
  Our objective is to define a disambiguation solution framework suitable for the task of WSD and NED employing the knowledge of semantic similarity derived from KGs. The formalization of the solution should put impact on unsupervised approach which can be used in various of domains and mainly exploit knowledge stored in KGs.

- **(3) Develop a similarity-based classification framework**.
  Our objective is to define a solution to allow concept classification systems to work in the heterogeneous environment of different KGs. Using similarity scores as features to train concept classifiers can obtain a general classifier containing similarity patterns between input words and concept's feature words. We aim to develop a similarity-based concept classification framework that only depends on feature words and similarity models, including both supervised and unsupervised classification approaches.

- **(4) Develop a semantic entity search framework for KGs**.
  Our objective is to develop a semantic entity search framework to search entities from KGs using natural language key word queries. The framework should map natural language queries to correct semantic resources contained in a given KG, and automatically formulate the mapped resources into proper SPARQL queries for retrieving entities from the KG. The query formulation engine should address the concept expansion problem in defining the SPARQL query.

## 1.3 Solution Outline

In order to fulfill the stated objectives, we propose a number of solutions put together in a single framework Sematch to facilitate their adoption. This framework aims to provide useful tools for the development of knowledge-based systems. Similarity computation is a key module employing various features of concepts and entities in KGs. Then similarity is used to develop unsupervised disambiguation method for words and entities. Furthermore, similarity is used as feature to substitute BOW representation for concept classification. Finally, similarity is used to develop a semantic search framework for mapping and discriminating

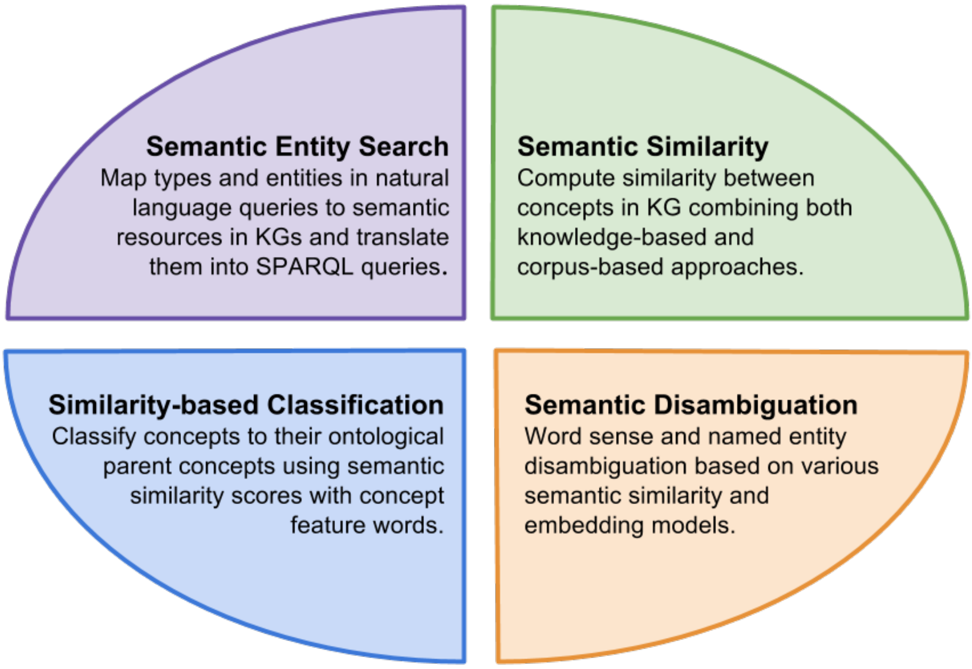

Semantic Entity Search
Map types and entities in natural language queries to semantic resources in KGs and translate them into SPARQL queries.
Semantic Similarity
Compute similarity between concepts in KG combining both knowledge-based and corpus-based approaches.
Similarity-based Classification
Classify concepts to their ontological parent concepts using semantic similarity scores with concept feature words.
Semantic Disambiguation
Word sense and named entity disambiguation based on various semantic similarity and embedding models.

Figure 1.4: Overview of thesis solutions scope and contributions.

natural language queries into KG concepts and entities. We also propose a rule-based approach for semantic parsing of keyword queries into SPARQL queries and implement a execution engine to retrieve entities from KGs. The proposed Sematch framework consists of five main elements (see Figure 1.4).

- **(1) Semantic similarity between concepts in KGs**.
  We propose a semantic similarity framework (**Objective 1**) for concepts in KGs. The framework generalizes the semantic information needed for computing semantic similarity. We identify the drawbacks of existing knowledge-based similarity methods and propose a new similarity method combining structure-based similarity method and information-based similarity method. In order to compute IC conveniently in KGs, we propose a novel graph-based IC computation method. In consequence, the information-based similarity methods as well as the proposed method are only dependent on the semantic information contained in KGs, instead of requiring a concept-annotated corpus to compute IC like the conventional corpus-based IC computation. This contribution is described in Chapter 3.

- **(2) Semantic disambiguation framework for KGs**.
  We propose a similarity based disambiguation framework to carry out the automatic disambiguation tasks handling the polysemous words and named entities (**Objective**

**2**). For the WSD task, we propose a Synset2Vec model for training word and synset embedding jointly only use the WordNet and its annotation corpus. Then context-based and graph-based disambiguation methods are used to implement disambiguation system based on the similarity between synsets and words in shared vector space. For the NED task, we propose to use word similarity and a Category2Vec model for discriminating polysemous entities. The Category2Vec model follows the similar idea of Synset2Vec, while the entity category and entity description are combined together to train the Category2Vec model. This contribution is described in Chapter 4.

- **(3) Concept classification and semantic entity search**.
  We propose two applications that rely on the similarity and disambiguation framework (**Objective 3 and Objective 4**). The similarity-based concept classification system uses a similarity model as feature representation to develop concept classifiers for ontological concepts. The semantic entity search system uses a similarity model and disambiguation model to develop entity search system over KG with natural language keyword queries. Thus, the entities can be retrieved through executing SPARQL queries in KG management systems. The two contributions are described in Chapter 5.

## 1.4 Thesis Organization

This dissertation is organized as follows. Chapter 2 describes the background of the thesis including semantic web, knowledge graph, and semantic similarity methods, as well as the illustration of their state of the art approaches. Chapter 3 describes the semantic similarity framework and Chapter 4 describes the semantic disambiguation framework. Chapter 5 presents the application of similarity-based concept classification and semantic entity search over KG. Finally, Chapter 6 concludes the thesis summarizing the contributions and proposing possible future researches to continue this work.

CHAPTER 2

# Foundations: Knowledge Graph and Similarity

*In terms of giving an extensive introduction to our research achievements, we describe the theoretical background of the thesis. The goal of this chapter is to give readers with enough theoretical background of the topics mentioned throughout the thesis, and to present the state of the art approaches in those research problems to which the thesis contributes novel solutions.*

*Firstly, we introduce semantic web and KG, especially their enabling technology stacks. Secondly, we describe the formal definition and classification of similarity methods, and then present the state of the art knowledge-based and corpus-based methods in measuring semantic similarity.*

## 2.1 Semantic Web and Knowledge Graph

The Semantic Web is defined as "a common framework that allows data to be shared and reused across application, enterprise, and community boundaries" according to the W3C[1], whose vision is to extend the current World Wide Web to bring a common structure for the web containing both machine-understandable and human-understandable data. With such goal, the Web of data described with HTML should contain meaningful descriptions so that both human and machine could understand. In order to provide more information about the meaning or relationships of web resources, the core research effort of the Semantic Web is related to modeling descriptive meta data about the web resources to facilitate the data publishing, consuming and storing. As the advancement of the development of Semantic Web, lexical resources to provide fine-grained meaning of the Web resource, have evolved from early efforts of folksonomies (e.g. flickr), schema (e.g. schema.org) to recent formal ontologies. The advantage of using a formal ontology is enabling agents to understand the content of Web resources and their semantics like human. The new representation of content resources can help to discover new knowledge of resources, which enables a reasoning ability through analysis the metadata and their semantics.

Moreover, the increasing popularity of Semantic Web and tremendous Web of data published online publicly with community efforts, have given birth to large scale KGs that are built under the hood of the Semantic Web technologies. Modern KGs contain millions of factual entities and their relationships which are described with well defined ontologies. KGs have inherited the key idea of the Semantic Web that large amount of knowledge are stored in KGs in a way that both human and computer agents can understand. Moreover, KGs have provided new opportunities to information management in answering more complex queries and reasoning the relationships between things.

With relation to this thesis, Semantic Web technologies provide fundamental data representation and information querying tool. For this reason, to introduce the reader to the concepts used further in the thesis, an overview of basic semantic web technologies are presented in the Section 2.1.1. Regarding KGs, as they are the fundamental data layer, we describe their key concepts and introduce widely used commonsense and encyclopedic KGs as well as popular KG-based applications in Section 2.1.2.

[1] https://www.w3.org/2001/sw/

### 2.1.1 RDF, RDFS, OWL and SPARQL

The semantic web technology stack proposed by Tim Berners-Lee et al. (2001) contains a number of layers. This section outines the key elements that are used in the thesis including RDF, Resource Description Framework Schema (RDFS), Ontology Web Language (OWL) and SPARQL.

RDF is a mark-up language built on top of Internationalized Resource Identifiers (IRI)[2] which is a generalization of URIs[3]. RDF provides an important data representational model and syntax for describing Web resources and their relationships. The underlying data structure of RDF is a labeled directed graph whose syntactic construct consists of triple statements including three components *subject*, *object* and *predicate* (Horrocks, 2008). A triple statement specifies a single edge (predicate) connecting two nodes (subject and object). For example, in a triple statement *dbr:Don_ Quixote, dbp:author, dbr:Miguel_ de_ Cervantes*, *dbr:Don_ Quixote* is the subject, *dbp:author* is the predicate, and *dbr:Miguel_ de_ Cervantes* is the object. Each RDF resource can be described by a number of *predicates* whose values are expressed by the *objects.* The *predicate* may be unary or binary. Specifically, the unary predicates connect with value objects (e.g. number or literal string), while binary predicates point to another resources. As illustrated in Table 2.1, the *rdfs:label* and *rdf:type* are unary predicates indicating the name and characteristic of entity *dbr:Don_ Quixote.* The authorship relation between the entity *dbr:Don_ Quixote* and the entity *dbr:Miguel_ de_ Cervantes* is a binary predicate. The advantage of RDF triple statements is the interoperability across systems in extending and integrating common RDF resources. Triple statements store the knowledge of semantic resources and can be perceived as a graph, where *subjects* and *objects* of RDF statements represent nodes of graph, while *predicates* denote edges.

A small subgraph of DBpedia related to *dbr:Don_ Quixote* is shown in Figure 1.2, which shows that the blue resources and green resources are connected by a special predicate *rdf:type.* It gives KGs capability of defining meanings to certain resources, such as the triple *dbr:Don_ Quixote, rdf:type , dbo:Book.* Specifically, the *rdf:type* denotes the class-instance relationship to represent the knowledge that *dbr:Don_ Quixote* is an instance of book. The term "book" is a special word that is able to express the abstraction of real world entities. Such abstract terms are usually defined as concepts in ontologies in order to provide well-defined meaning to identify and distinguish entities. In computer science, an ontology is a model of the world that introduces vocabulary describing various aspects of the domain being modeled and provides explicit specification of the intended meaning of that

[2] https://www.w3.org/International/

[3] http://www.ietf.org/rfc/rfc3986

| Subject | Predicate | Object |
|---|---|---|
| dbr:Don_Quixote | rdfs:label | "Don Quixote" |
| dbr:Don_Quixote | rdf:type | dbo:Book |
| dbr:Don_Quixote | dbp:author | dbr:Miguel_de_Cervantes |
| dbr:Spain | dbp:capital | dbr:Madrid |
| dbr:Madrid | rdf:type | yago:City108524735 |
| dbr:Spain | rdf:type | dbo:Country |

Table 2.1: DBpedia Triples about Don Quixote, Madrid and Spain

vocabulary (Horrocks, 2008). Moreover, the specification often includes a concept taxonomy to distinguish various conceptual features, such as singers are artists. RDFS is a basic RDF vocabulary description language that extends RDF and consists of several resources to define concepts in ontology such as *rdfs:Class* and *rdfs:subClassOf.* For example, the concept *singer* and *artist* can be defined as a *rdfs:Class*, while their hierarchical relations can be represented by the predicate *rdfs:subClassOf* in a RDF triple *singer, rdfs:subClassOf, artist.* In fact, due to the commonality of concept taxonomies in KGs, in order to represent both super-concept and sub-concept relation, Simple Knowledge Organization System (SKOS)[4] is usually used to describe large scale concept taxonomy, such as Wikipedia category in DBpedia.

Moreover, as RDFS can only define ontologies with very limited elements, OWL (McGuinness et al., 2004) becomes a de-facto ontology language standard (Horrocks, 2008) of KGs in order to express various relationships between semantic resources with more details, such as *dbp:author, dbp:capital,* and other logical features. OWL is fundamentally built on top of Description Logics (DL) (Baader, 2003) consisting of logic-based knowledge-representation formalism which is described in terms of instances, concepts and properties. Instances corresponds to entities (such as *dbr:Don_ Quixote*), concepts (also called "classes" in RDF such as *dbo:Book*) describe sets of instances sharing similar characteristics, and properties specify relationships between concepts and instances. In consequence, OWL is able to provide logical expressions, local properties, and to define certain domain and range for predicates. In addition, description capabilities such as constructs (e.g. union, intersection) and axioms (e.g. subclass and equivalent class) are also available in OWL.

[4] https://www.w3.org/TR/swbp-skos-core-spec

```
PREFIX owl:<http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX yago: <http://dbpedia.org/class/yago/>
SELECT ?singer WHERE {
        ?singer rdf:type yago:Singer110599806 .
        ?singer dbp:nationality ?country .
        dbr:Don_Quixote dbp:author ?writer .
        ?writer dbp:nationality ?country .
}
```

Table 2.2: Examples of the usage of SPARQL to retrieve a list of singers.

In addition to the languages for describing the resources and defining meanings of metada, SPARQL[5] is a W3C recommendation of the RDF query language which can query and manipulate data stored in RDF. Thus, this is also a semantic query language for retrieving entity-centric information from RDF-based KG. Furthermore, SPARQL enables to formulate queries with triple patterns like triple statement stored in RDF database. Therefore, triple patterns can be viewed as graph patterns which can be executed as graph pattern matching in the specific database. Graph pattern matching can answer more complex queries to infer information based on the given triple patterns. For example, to answer the query "singers from the same country than the author of Don Quixote", the SPARQL query shown in Table 2.2 can be used to return the proper singers. Although we do not specify who is the the author of the book Don Quixote, the SPARQL query construct a query graph to infer the author as a intermediate node in the graph query, therefore, the execution of the SPARQL query can return a list of singers having the same nationality of the writer of Don Quixote.

### 2.1.2 Knowledge Graph Formalization

With the increasing popularity of the Semantic Web initiative, many public KGs have become available, such as Freebase (Bollacker et al., 2008), DBpedia (Bizer et al., 2009b), and YAGO (Hoffart et al., 2013), which are novel semantic networks recording millions of

[5] https://www.w3.org/TR/sparql11-query/

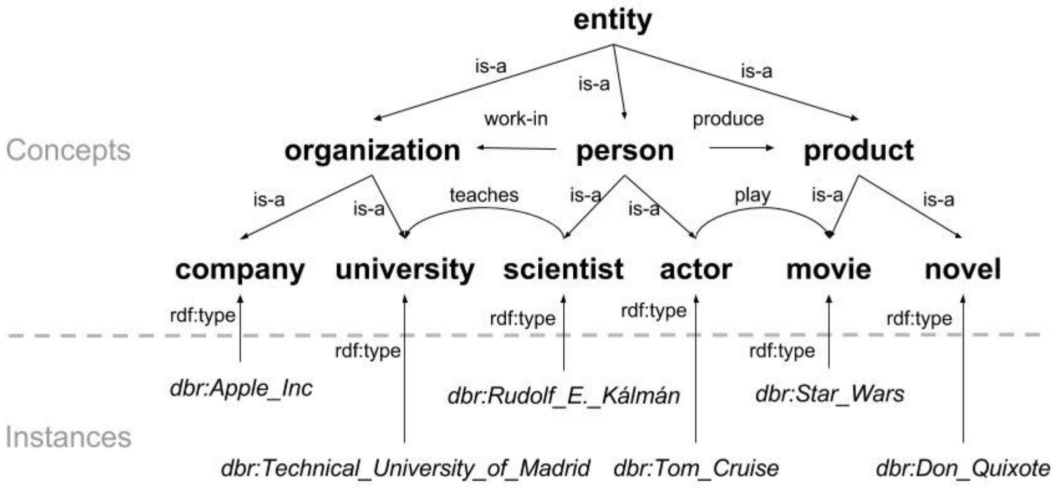


Figure 2.1: A Tiny Example of Knowledge Graph

| Entity | Type | Concept |
|---|---|---|
| dbr:Star_Wars | yago:Movie106613686, dbo:Film | Movie |
| dbr:Don_Quixote | yago:Novel106367879, dbo:Book | Novel |
| dbr:Tom_Cruise | yago:Actor109765278, dbo:Actor | Actor |
| dbr:Apple_Inc | yago:Company108058098, dbo:Company | Company |

Table 2.3: The Examples of Mapped Entities and Entity Types in DBpedia.

concepts, entities and their relationships. Those public KGs take advantage from public encyclopedia knowledge bases such as Wikipedia, as well as advanced information extraction techniques (Banko et al., 2007) and knowledge population techniques (Ji and Grishman, 2011). Formally, nodes of KGs consist of a set of concepts $C_1, C_2, \ldots, C_n$ representing conceptual abstractions of things (e.g. dbo:Book), and a set of instances $I_1, I_2, \ldots, I_m$ representing real world entities (e.g. dbr:Don_Quixote), while edges of KGs denote the semantic relation between entities (e.g. dbp:author ). Following DL terminology (Horrocks, 2008), a KG contains two types of axioms: a set of axioms is called a terminology box (TBox) that describes constraints on the structure of the domain, similar to the conceptual schema in database setting, and a set of axioms is called assertion box (ABox) that asserts facts about concrete situations, like data in a database setting (Horrocks, 2008). Concepts of the KG contain axioms describing concept hierarchies and are usually refereed as ontology classes (TBox), while axioms about entity instances are usually referred as ontology instances (ABox). Both TBox and ABox of KGs provide rich semantic information for KG-based applications. Figure 2.1 shows an example of a KG using the above notions. Concepts of TBox are constructed hierarchically and classify entity instances into different types (e.g., actor or movie) through a special semantic relation *rdf:type*(e.g., dbr:Star_Wars is an instance of concept movie). Concepts and hierarchical relations (e.g., is-a) compose a concept taxonomy which is a concept tree where nodes denote the concepts and edges denote the hierarchical relations. The hierarchical relations between concepts specify that a concept $C_i$ is a kind of concept $C_j$ (e.g., *actor* is a *person*). Apart from hierarchical relationships, concepts can have other semantic relationships among them (e.g., *actor* plays in a *movie*). Note that the example KG is a simplified example from DBpedia for illustration, and Table 2.3 shows examples of DBpedia entities and their types which are mapped to the example KG in Figure 2.1.

A special semantic information of TBox is that it gives hierarchical categorizations of entities. TBox can be conceptualized as a concept taxonomy which is very similar to the characteristics of the lexical database WordNet (Miller, 1995) which has been conceptualized as a conventional semantic network of the lexicon of English words. WordNet can be viewed as a concept taxonomy where nodes denote WordNet synsets representing a set of words that share one common sense (synonyms), and edges denote hierarchical relations of hypernym and hyponymy (the relation between a sub-concept and a super-concept) between synsets. Recent efforts have transformed WordNet to be accessed and applied as a concept taxonomy in KGs by converting the conventional representation of WordNet into a novel linked data representation. For example, KGs such as DBpedia, YAGO and BabelNet (Navigli and Ponzetto, 2012) have integrated WordNet and used it as part of a concept taxonomy to categorize entity instances into different types. For example, *yago:Actor109765278* is an example of integrating WordNet in YAGO and used in DBpedia.

Such integration of conventional lexical resources and novel KGs has provided novel opportunities to facilitate many different NLP and IR tasks (Hovy et al., 2013), including WSD (Navigli, 2009; Moro et al., 2014), NED (Hoffart et al., 2012a; Hulpus et al., 2015), query interpretation (Pound et al., 2010a), document modeling (Schuhmacher and Ponzetto, 2014) and question answering (Shekarpour et al., 2015) to name a few. Those KG-based applications rely on the knowledge of concepts, instances and their relationships. In this thesis, we mainly exploit the semantic information contained in KGs and use similarity to encode those knowledge for developing KG-based applications.

## 2.2 Semantic Similarity

Measuring similarity or distance between two data objects is a key module in data mining and knowledge discovery that involves distance computation such as clustering, information retrieval, recommendation and classification. For continuous numerical data, the popular distance metrics Minkowski Distance of order one (Manhattan) and order two (Euclidean) are widely used as distance metrics (Aggarwal, 2003). However, regarding to KG-based applications, similarity or distance metrics are needed for categorical data, such as concepts (TBox), entities (ABox), words (literal values), whose similarity or distance computations are not straightforward comparing to numerical data. The simplest way to find similarity between two categorical data is to assign a similarity score of 1 if two values are identical and a similarity score of 0 if two values are not identical, which is commonly known as exact matching model (Belkin and Croft, 1992). Improvement to this simple model measures similarity between two categorical data using lexical matching approaches that produce similarity scores based on the number of lexical units occurring in both strings such as Levenshtein distance (Yujian and Bo, 2007). Further improvements have considered stemming, stop-word removal, part of speech tagging, longest subsequence matching, as well as various weighting and normalization factors (Mihalcea et al., 2006). However, those similarity or distance methods fail to identify the meaning of data ignoring those semantically similar but lexically different data such as,“movie” and “film”, “baritone” and “singer”.

We focus on semantic similarity to quantify how much two objects (e.g. concept, word or entity) are alike to each other respect to their meanings. In this thesis, semantic similarity is used as fundamental feature for disambiguation, classification and semantic search in KG. Therefore, in order to provide a thorough background of the topic, the following section gives formal definitions, terminology and approach classification of similarity, and summarizes the state of the art methods in measuring semantic similarity.

### 2.2.1 Distance and Similarity

Many similarity metrics are first defined as semantic distances and then converted to similarity metrics. Thus, we define the function property of distance and similarity. Given two objects $A$ and $B$ which can be concepts, words, texts and entities, we define the semantic distance function $Distance(A, B)$ whose properties satisfy the following axioms:

**Minimal Property:** $Distance(A, B) \geq Distance(A, A) = 0$.

**Symmetric Property:** $Distance(A, B) = Distance(B, A)$.

**Triangle inequality:** $Distance(A, B) + Distance(B, C) \geq Distance(A, C)$.

The minimal axiom implies that the distance between an object and itself is the same for all objects. Similarly, the $Similarity(A, B)$ function should satisfy the following axioms:

**Maximal Property:** $Similarity(A, B) \leq Similarity(A, A) = 1$.

**Symmetric Property:** $Similarity(A, B) = Similarity(B, A)$.

**Triangle inequality:** $Similarity(A, B) + Similarity(B, C) \leq Similarity(A, C)$.

For convenient comparison among metrics in evaluation, we only consider similarity between objects, thus, those distance metrics need to be converted into similarity metrics. There are three transformation function that are commonly used to transform distance scores to similarity scores, which are shown in the following functions.

$$Similarity(A, B) = 1 - Distance(A, B) \tag{2.1}$$

$$Similarity(A, B) = \frac{1}{1 \quad + \quad Distance(A, B)} \tag{2.2}$$

$$Similarity(A, B) = \log_2 (1 \quad + \quad Distance(A, B)) \tag{2.3}$$

When the $Distance(A, B) \in [0, 1]$, all three transformation functions shown above will give the score of $Similarity(A, B)$ ranged in $[0, 1]$ that satisfy the maximal property of similarity function. However, when $Distance(A, B) > 1$, the Equation 2.1 and Equation 2.3 are not able to guarantee the similarity score ranged in $[0, 1]$. This property should be awared while choosing the specific transformation function. Because the similarity score of Equation 2.2 is always ranged in $(0, 1]$ which always satisfy the maximal property if the distance satisfy the minimal property, we use it to derive similarity metrics in this thesis.

### 2.2.2 Terminology and Classification

The problem of formalizing and quantifying the intuitive notion of similarity has a long history in philosophy, psychology, and Artificial Intelligence (AI), and many different perspectives have been suggested (Budanitsky and Hirst, 2001). We first distinguish some terminologies and properties in order to present the similarity topic clearly and minimize the ambiguity in using the terms *semantic similarity* and *semantic relatedness*.

The similarity between concepts, words and entities can be generally categorized into *relational similarity* and *attributional similarity* according to cognitive science (Gentner, 1983). Take words as example, the *attributional similarity* measures the degree of correspondence between the properties of word $a$ and $b$ (e.g. black and white are both kinds of color, loud and quiet are both kinds of sound), while the *relational similarity* measures two pairs of words $a : b$ and $c : d$ based on the degree of correspondence between the relations of word pairs $a : b$ and $c : d$ (Turney et al., 2010). For example, “dog” and “wolf” have a relatively high degree of *attributional similarity*, whereas “dog:bark” and “cat:meow” have a relatively high degree of *relational similarity* (Turney, 2006). In this thesis and the following review of similarity methods, only *attributional similarity* is considered, thus when we mention the term *similarity* we refer to *attributional similarity* between two objects such as concepts, words or entities.

In computational linguistics, semantic relatedness is inverse of semantic distances (Budanitsky and Hirst, 2001) and corresponds to attributional similarity in cognitive science (Gentner, 1983). Semantic relatedness assumes that two objects are semantically related if they have any kind of semantic relations (Budanitsky and Hirst, 2001). Semantic similarity is a special metric that represents the commonality of two concepts relying on their hierarchical relations (Resnik, 1995; Budanitsky and Hirst, 2001; Turney et al., 2010). In general, semantic similarity, also called taxonomical similarity, is a specific type of attributional similarity (Turney et al., 2010), thus it is a special case of semantic relatedness which is a more general concept and does not necessarily rely on hierarchical relations. For example, in case of WordNet, semantic similarity mainly considers hypernym and hyponym relations (super-concept and sub-concept in taxonomy), such as “car” and “bicycle” are semantically similar because they share the hypernym “vehicle” (Resnik, 1995). On the other hand, semantic relatedness also considers meronyms (“car” and “wheel”) and antonyms (“hot” and ”cold"), which are functionally related or frequently associated (“pencil and paper”) (Turney et al., 2010). Note that antonyms have a high degree of attributional similarity such that hot and cold are kinds of temperature; black and white are kinds of color; loud and quiet are kinds of sound (Turney et al., 2010). Moreover, in case of encyclopedic KGs such as the example shown in Figure 2.1, *scientist* and *actor* are semantically similar because they share the super-concept *person*. Although *actor* and *movie* are clearly related, but they are not really similar because they belong to different branches of taxonomy.

There is a relatively large number of similarity metrics which were previously proposed in the literature. Among them, there are mainly two types of approaches, namely knowledge-based approaches and corpus-based approaches (Mihalcea et al., 2006). Knowledge-based approaches are also called ontology-based approaches because they rely on the knowledge contained in semantic networks or formal ontology such as structural knowledge of concept taxonomy (e.g. depth and path length) and statistical information derived from concept-annotated corpus. On the other hand, corpus-based approaches are based on models of distributional similarity (Harris, 1954) learned from large text collections relying on data distributions. Note that with the term "data" we refer to concepts, words or entities. Two data objects will have a high distributional similarity if their surrounding contexts are similar. Since only the occurrences of data are counted without identifying the specific meaning of data and detecting the relations between data, corpus-based approaches consider all kinds of relations and mainly measure semantic relatedness. In comparison, knowledge-based approaches usually measure semantic similarity as they mainly use the semantic information in ontologies to define similarity metrics. In the following chapters of this thesis, we do not specially differentiate semantic similarity and semantic relatedness. However, when we say knowledge-based similarity, we mainly refer to semantic similarity, while the corpus-based similarity is referred to general semantic relatedness denoting the general attributional similarity involving meaning. Corpus-based similarity methods have wider computational applications because they consider all kinds of semantic relations between data. Knowledge-based similarity methods would be more useful when applications need to encode hierarchical relations between concepts, such as concept expansion and concept-based retrieval (Dragoni et al., 2012). In general, semantic similarity metrics can be used for weighting or ranking similar concepts based on a concept taxonomy. In such way, semantic similarity methods could be applied in KGs for concept-based entity retrieval or QA, where those entities that contain types having similar meaning to query concepts would be retrieved. Furthermore, in entity modeling, semantic similarity could be used to cluster entities based on their concepts. In the following sections, we present the state of art of knowledge-based similarity methods and corpus-based similarity methods respectively.

### 2.2.3 Knowledge-based Similarity Methods

Several methods for determining similarity between concepts represented in an ontology (e.g. Gene Ontology, WordNet, UMLS and MeSH), or in schemas (XML or Database ) have been proposed and applied in wide range of domains. Most of knowledge-based semantic similarity metrics are reported having good performance in measuring the semantic similarity between concepts in WordNet and Gene Ontology. Because the semantic information they used to define similarity metrics are based on an ontology, those metrics are still applicable to measure similarity in ontology of KGs, such as DBpedia, YAGO, BabelNet, especially for their concept taxonomy. An ontology can be defined as a directed labeled graph, $G = (V, E, \tau)$, where $V$ is a set of nodes, $E$ is a set of edges connecting those nodes; and $\tau$ is a function $V \times V \rightarrow E$ that defines all triples in $G$. Knowledge-based similarity methods measure the similarity between concepts $c_1, c_2 \in V$, formally $sim(c_1, c_2)$, using semantic information contained in $G$. In this section, we present the state of the art of knowledge-based similarity methods in three categories according to their properties (Sánchez et al., 2012): (1) based on how close two concepts in the taxonomy are, structure-based methods; (2) based on how much information two concepts share, information-based methods; (3) based on the properties of the concepts, feature-based methods.

#### 2.2.3.1 Structure-based Similarity Methods

Structure-based methods, which are also called edge-based or hierarchical methods, use the edges counts (shortest path length) and edge types (depth) as the information source for defining similarity metrics. Ontology classes can be conceptualized as a concept taxonomy and viewed as a directed graph in which concepts are interrelated mainly by means of taxonomic (IS-A) relations. They are called structure-based methods because the similarity between two concepts $c_i$, $c_j$ is usually measured by determining the path linking them or their positions in the taxonomy. Most existing structure-based methods proposed in literature compute the similarity according to the shortest path length between concepts and concept depth. Because they rely on the structure of the ontology to compute semantic similarity, the computation of similarity is much simpler and has lower computational complexity. However, the performance of structure-based methods relies on well defined ontologies since the similarity scores would be fixed when the ontology and similarity methods are established. This section investigates the current state of the art of structure-based methods separately in details.

Rada et al. (1989) hypothesized that when only is-a relations are used in semantic nets, semantic relatedness and semantic distance are equivalent. Thus, they proposed to measure the distance between concepts represented in hierarchical taxonomy in order to properly rank the documents in response to a query. Semantic distance is defined as the metric by counting the shortest path length between two concepts. A path $P(c_i, c_j)$ between $c_i, c_j \in V$ through $G$ is a sequence of nodes and edges $P(c_i, c_j) = \{c_i, e_i, \dots, v_k, e_k, v_{k+1}, e_{k+1}, \dots, c_j\}$ connecting the concepts $c_i$ and $c_j$ with cardinality or size $n$. For every two consecutive nodes $v_k, v_{k+1} \in V$ in $P(c_i, c_j)$, there exists an edge $e_k \in E$. Note that though $G$ is modeled as a directed graph we do not consider the direction of edges because semantic relations can be considered to have semantically sound inverse relation (Hulpus et al., 2015). Let $Paths(c_i, c_j) = \{P_1, P_2, \dots P_n\}$ be the set of paths connecting the concepts $c_i$ and $c_j$ with cardinality or size $N$. Let $|P_i|$ denote the length of a path $P_i \in Paths(c_i, c_j)$, then $length(c_i, c_j) = \underset{1 \leq i \leq N}{min}(|P_i|)$ denotes the shortest path length between two concepts. Rada et al. (1989) proposed to use the shortest path length between concepts to represent their semantic distance as expressed in Eq.(2.4):

$$Distance_{Rada}(c_i, c_j) = length(c_i, c_j) \tag{2.4}$$

Semantic distance is the most intuitive semantic information and the shorter the path from one concept to another, the more similar they are. In order to transform semantic distance into semantic similarity, the distance function is transformed into the **path** similarity methods as expressed in Eq.(4.11):

$$sim_{path}(c_i, c_j) = \frac{1}{1 + length(c_i, c_j)} \tag{2.5}$$

The **lch** (Leacock and Chodorow, 1998) method measures the semantic similarity between concepts based on their shortest path length using a non-linear function illustrated in Eq.(2.6):

$$sim_{lch}(c_i, c_j) = -\log\left(\frac{length(c_i, c_j)}{2 * D}\right) \tag{2.6}$$

where $D$ is the maximum depth of the concept taxonomy. Hirst and St-Onge (1998) extended the path counting by considering various types of relations used in the WordNet. It considers relation categories: upward (hypernymy and meronymy), downward (such as hyponymy and holonymy) and horizontal (such as antonymy). Both path length $length(c_i, c_j)$, and number of changes of type of edge, $change(c_i, c_j)$ are used to define the **h&s** similarity function:

$$sim_{h\&s}(c_i, c_j) = C - length(c_i, c_j) - k \times change(c_i, c_j) \tag{2.7}$$

where $C$ and $k$ are constant parameters ($C = 8$ and $k = 1$ are used by the authors), and $change(c_i, c_j)$ is the number of times the edges change. Since various of semantic relations are employed in the **h&s** method apart from taxonomical relations, it captures more general semantic similarity.

In addition to the path length between concepts, it is intuitive to employ concept depth information in taxonomy because concepts at upper layers of the hierarchy have more general semantics and less similarity between them, while concepts at lower layers have more concrete semantics and stronger similarity. Thus, the idea of using depth information of concepts to measure the semantic similarity lies in the property of concept taxonomies that the upper-level concepts in a taxonomy are supposed to be more general. In consequence, the similarity between lower-level concepts should be considered more similar than those concepts between upper-level concepts. For example in Figure 2.1, the concept pair *scientist* and *actor* are more similar than the concept pair *person* and *product*. We define the path between the root concept and a given concept through hierarchical relations as depth. Formally, the $depth(c_i) = length(c_i, c_{root})$ of a concept $c_i \in V$ is defined as the shortest path length from $c_i$ to root concept $c_{root} \in V$. For every two consecutive nodes $v_k, v_{k+1} \in P(c_i, c_{root})$, there exists an edge $e_k \in \{hypernym, subClassOf\}$. Furthermore, we define a special concept Least Common Subsumer (LCS) which is the most specific concept that is a shared ancestor of the two concepts. It can represent the common characteristic shared by two sub-concepts. For example, the LCS of concept *scientist* and concept *actor* is the concept *person*. Let $c_{lcs}$ be the LCS of concepts $c_i$ and $c_j$, the **wup** (Wu and Palmer, 1994) method measures semantic similarity of given concepts considering relative depth of concepts with the following formula:

$$sim_{wup}(c_i, c_j) = \frac{2depth(c_{lcs})}{depth(c_i) + depth(c_j)} \tag{2.8}$$

Similarly, the **li** method (Li et al., 2003) combines the shortest path length and the depth of LCS. It measures semantic similarity using a non-linear functions as shown in Eq.(2.9).

$$sim_{li}(c_i, c_j) = e^{-\alpha length(c_i, c_j)} \cdot \frac{e^{\beta depth(c_{lcs})} - e^{-\beta depth(c_{lcs})}}{e^{\beta depth(c_{lcs})} + e^{-\beta depth(c_{lcs})}} \tag{2.9}$$

where $e$ is the Euler's number and $\alpha, \beta$ are parameters that contribute to the path length and depth respectively. According to the experiment of **li** (Li et al., 2003), the empirical optimal parameters are $\alpha = 0.2$ and $\beta = 0.6$. Note that the optimal parameters are just empirical results in a specific setting which is lack of theoretical foundations and is not able to be generalized.

In summary, structure-based similarity methods combine several structural knowledge of ontology such as path length, depth and LCS. The main advantage is the computational simplicity and convenient adaption to new domain ontologies since they only depend on the knowledge derived from ontology. However, they assume the uniform distance between concepts (path length and depth are based on count of edges) which would be influenced by the quality of the ontology in terms of granularity degree and concept details of taxonomy.

#### 2.2.3.2 Information-based Similarity Methods

Some knowledge-based semantic similarity methods (Resnik, 1999; Lin, 1998; Jiang and Conrath, 1997) leverage IC of concepts to improve performance of measuring semantic similarity in structure-based similarity methods. The definition of IC was proposed and introduced by  Resnik (1995) which is computed from the information distribution of concepts over the concept-annotated corpora. Formally, the IC of concepts are computed according to the negative log of their probability of occurrence in a given corpus. The $IC_{corpus}(c_i)$ of a concept $c_i \in V$ is defined as: $IC_{corpus}(c_i) = -logProb(c_i)$, where $Prob(c_i)$ denotes the probability of encountering the set of $words(c_i)$ subsumed by concept $c_i$. Let

$freq_{corpus}(c_i) = \sum_{w \in words(c_i)} count(w)$ be the frequency of concept $c_i$ occurs in corpus, then $Prob(c_i) = \frac{freq_{corpus}(c_i)}{N}$ where $N$ is the total number of concepts observed in corpus. With this definition of IC, the infrequent concepts are considered more informative than common ones.

Specifically, the quantitative characteristic of IC is that the more abstract concepts have lower value of IC and more specific concepts have higher value of IC. If two concepts share a more specific concept, it means that they share more information and thus more similar because the IC of their LCS is higher. Based on this intuition, Resnik (1995) proposed the **res** method based on the amount of shared information between two concepts, represented by their LCS which is the most specific concept that is an ancestor of two concepts. Thus, the similarity between concepts is computed as the IC of their LCS which is illustrated in Eq.(2.10).

$$sim_{res}(c_i, c_j) = IC_{corpus}(c_{lcs}) \tag{2.10}$$

Since any pair of concepts having the same LCS results in the same semantic similarity, Lin (1998) extended Resnik's work by computing the similarity between concepts as the ratio between the IC of LCS and their own ICs. The **lin** (Lin, 1998) method is defined as:

$$sim_{lin}(c_i, c_j) = \frac{2IC_{corpus}(c_{lcs})}{IC_{corpus}(c_i) + IC_{corpus}(c_j)} \tag{2.11}$$

Similarly, the **jcn** (Jiang and Conrath, 1997) method measures the difference between concepts by subtracting the sum of the IC of each concept alone from the IC of their LCS.

$$dis_{jcn}(c_i, c_j) = IC_{corpus}(c_i) + IC_{corpus}(c_j) - 2IC_{corpus}(c_{lcs}) \tag{2.12}$$

It can be transformed from distance $dis_{jcn}(c_i, c_j)$ to similarity $sim_{jcn}(c_i, c_j)$ by computing the reverse of distance:

$$sim_{jcn}(c_i, c_j) = \frac{1}{1 + dis_{jcn}(c_i, c_j)} \tag{2.13}$$

Lastra-Díaz and García-Serrano (2015a) further proposed to use cosine-normalized **jcn** similarity method with conditional probabilities and IC for weighting the shortest path between concepts, in order to make the metric suitable for any type of taxonomy.

The computation of corpus-based IC introduced above requires concept-annotation corpus which is dependent on proper disambiguation of concepts or highly cost manual annotation. Moreover, the corpus should be selected considering the proper distribution of concepts. These limitations prevent the applications for corpus-based IC to new domains. Due to those inconveniences, ontology-based IC computation methods are proposed to overcome those limitations. From information theory view, abstract concepts appear more probably in computing corpus-based IC because they subsume many sub-concepts. With such consideration, the probability of appearance of concepts can be estimated as a function of counting the number of their hyponyms in a taxonomy. Seco et al. (2004) proposed intrinsic-based IC only based on the structure of the ontology. Let $hypo(c)$ denote the number of hyponyms of the concept $c$, and $max.nodes$ denote total number of concepts in a taxonomy, the intrinsic-based IC is defined as:

$$IC_{intrinsic} = 1 - \frac{\log(hypo(c) + 1)}{\log(max.nodes)} \tag{2.14}$$

where the denominator ensures that $IC_{intrinsic}$ value is normalized in the range of $[0, 1]$. The intrinsic-based IC reaches maximal value when a concept has no sub-concepts (e.g. leaf concept). As intrinsic-based IC only relies on the number of sub-concepts, those concepts having different depth but the same number of hyponyms would have equal value of IC. In order to address this drawback, the concept depth is combined with hyponym and results in dIntrinsic-based IC (Zhou et al., 2008):

$$IC_{dIntrinsic} = k(1 - \frac{\log(hypo(c) + 1)}{\log(max.nodes)}) + (1 - k)(\frac{\log(depth(c))}{\log(max.depth)}) \tag{2.15}$$

where $max.depth$ is the maximum depth of the taxonomy, while the factor $k$ adjusts the weight ($k = 0.5$ is used by authors). Moreover, in order to consider other relations connecting concepts in ontology (e.g. part-of), eIntrinsic-based IC (Pirró and Euzenat, 2010) is proposed to take into consideration of whole set of relations connecting a concept $c$ with other concepts:

$$IC_{Eintrinsic} = \sum_{j=1}^{m} \frac{\sum_{k=1}^{n} IC_{intrinsic}(c_k \in C_{R_j})}{|C_{R_j}|} \tag{2.16}$$

which considers all the $m$ kinds of relations $R$. For each connected concept $c_k \in C_{R_j}$ the average intrinsic-based IC is computed. The final IC value of the concept $IC_{eIntrinsic}(c)$ is combined with intrinsic-based IC with weighting parameters:

$$IC_{eIntrinsic}(c) = \alpha IC_{Eintrinsic}(c) + \beta IC_{intrinsic}(c) \tag{2.17}$$

where parameter $\alpha$ and $\beta$ are used to give more or less emphasis to the hierarchical intrinsic-based IC of the concepts.

In general, information-based similarity methods require good calculation of IC which has higher computational complexity but can overcome the limitation of the pure structure-based methods which treat edges with uniform distance. For a more detailed survey and evaluation of information-based similarity methods on WordNet, readers can refer to (Lastra-Díaz and García-Serrano, 2015b).

#### 2.2.3.3 Feature-based Similarity Methods

Feature-based similarity methods take into account of both common and distinguish features of the objects being compared which are based on set theory and were first proposed by Tversky (Tversky, 1977). Generally, the features held in common increase the similarity and features not held in common decrease the similarity. Tversky (1977) proposed the feature-based similarity method based on the feature sets of concepts, such as the attributes associated with the concepts or the textual definitions of the concepts. In such a way, assessment of similarity is defined as a comparison of features rather than as the computation of metric distances between data points. The similarity between concept $c_i$ and $c_j$ is computed from the functions of features: (1) features common to $c_i$ and $c_j$; (2) features in $c_i$ but not in $c_j$; (3) features in $c_j$ but not in $c_i$. Common features tend to increase the similarity and non-common features tend to diminish the similarity of two concepts. Suppose that we use function $\Psi(c)$ to describe features of concept $c$, indicating its properties or descriptions, thus, the more common features and the less non-common features between $\Psi(c_i)$ and $\Psi(c_j)$, the more similar the two concepts are. A *contrast model* (Tversky, 1977) of similarity function has been proposed below.

$$sim_{t\&c}\left(c_i, c_j\right) = \theta F(\Psi(c_i) \cap \Psi(c_j)) - \alpha F(\Psi(c_i) - \Psi(c_j)) - \beta F(\Psi(c_j) - \Psi(c_j)) \tag{2.18}$$

where the $F$ is a non-negative interval scale function that represents the salience of a set of features, such as the cardinal of the feature set. The $\Psi(c_i) \cap \Psi(c_j)$ denotes the features that are common to both $c_i$ and $c_j$, while the $\Psi(c_i) - \Psi(c_j)$ and $\Psi(c_j) - \Psi(c_j)$ stands for the features that only belong to $c_i$ or $c_j$ respectively. The parameters $\theta, \alpha, \beta \geq 0$ define a family of scales providing different contributions on the different components. Since the similarity value of the above function is not normalized between 0 and 1, a *ratio model* (Tversky, 1977) has been proposed in order to be independent of the size of the features being compared and to bound the similarity scores in the range of $[0, 1]$:

$$sim_{Tversky}\left(c_i, c_j\right) = \frac{F(\Psi(c_i) \cap \Psi(c_j))}{F(\Psi(c_i) \cap \Psi(c_j)) + \alpha F(\Psi(c_i) - \Psi(c_j)) + \beta F(\Psi(c_j) - \Psi(c_j))}. \tag{2.19}$$

The $0 \leq \alpha, \beta \leq 1$ defines the relative importance of the common and non-common features. If $\alpha = \beta = 1$, the similarity method becomes as Jaccard coefficient:

$$sim_{jaccard}\left(c_i, c_j\right) = \frac{F(\Psi(c_i) \cap \Psi(c_j))}{F(\Psi(c_i) \cup \Psi(c_j))} \tag{2.20}$$

And if $\alpha = \beta = \frac{1}{2}$, the similarity method becomes dice coefficient:

$$Sim\left(A, B\right) = \frac{2F(\Psi(c_i) \cap \Psi(c_j))}{F(\Psi(c_i)) + F(\Psi(c_j))} \tag{2.21}$$

The formula 2.19 is based on the assumption that the similarity should not be a symmetric relation. Thus, the variables of $\alpha$ and $\beta$ provide a systematic approach to determine the asymmetry of the similarity evaluation. When $\alpha = \beta$, the similarity method is symmetric and not directional, thus it can assess the degree to which two concepts are similar to each other. Rodríguez (Rodríguez and Egenhofer, 2003) extended the similarity function that is defined by the weighted sum of the similarity of each specification component of synsets in WordNet. Sánchez et al. (Sánchez et al., 2012) extended Tversky's work by only considering taxonomical knowledge and using dissimilarity as their metric. Hossein et al. (Zadeh and Reformat, 2012) also proposed a variation of feature based method based on fuzzy set theory.

Furthermore, other works also consider structure-based methods and information-based methods as additional features and combine them together to derive a better performance similarity method. Pirró and Euzenat (2010) proposed to use IC for measuring common and non-common features:

$$sim_{FaITH}(c_i, c_j) = \frac{IC(c_{lcs})}{IC(c_{lcs}) + \alpha(IC(c_i) - IC(c_{lcs})) + \beta(IC(c_j) - IC(c_{lcs}))}. \tag{2.22}$$

where the IC of the LCS concept is used to represent the common feature, while the non-common features are derived from the individual concept's IC minus their LCS concept's IC. In order to make the similarity method symmetric, the $\alpha$ and $\beta$ are set to be 1. In consequence, the symmetric metric is defined as:

$$sim_{FaITH}(c_i, c_j) = \frac{IC(c_{lcs})}{IC(c_i) + IC(c_j) - IC(c_{lcs})}. \tag{2.23}$$

In summary, feature-based similarity methods are derived from the set theory and try to overcome the limitations of other methods by exploiting more semantic information and considering both commonalities and differences between concepts. As a result, feature based methods are more general and can be applied in the situation that the other methods can't be applied directly or combining multiple information sources for defining general semantic relatedness. However, the feature-based similarity methods are limited to the case that various features of concepts are available. Moreover, another problem is to determine the parameters giving different weights of features, which is difficult to decide automatically.

### 2.2.4 Corpus-based Similarity Methods

Corpus-based similarity methods measure the similarity between concepts or words based on their occurrence information gained from large corpora. The word occurrence can be directly counted from the given corpus, whereas concept occurrence can only be counted after the concept annotation of the given corpus either by human or automatic systems. As corpus-based methods are modeled based on textual corpora and distributional information that would make a wide variety of words to be considered as related, they usually measure the general semantic relatedness between words rather than the specific semantic similarity that depends on hierarchical relations (Turney et al., 2010). The basic idea of corpus-based similarity methods is based on word associations learned from large text collections following the distributional hypothesis (Harris, 1954), which is defined in linguistic perspective as "the words that occur in similar contexts tend to have similar meanings". Two words are assumed to be more similar if their surrounding contexts are more similar or they appear together more frequently. In this section, we present the state of art corpus-based similarity methods in three categories: (1) based on co-occurrence statistics; (2) based on semantic analysis; (3) based on the embedding approach.

#### 2.2.4.1 Statistical Co-occurrence

The most intuitive and widely-used corpus-based similarity methods are based on statistics of word distributions and word co-occurrences. The key idea of the statistics is using the word counts or the raw frequency of the words in textual corpus. Church and Hanks (1990) proposed Pointwise Mutual Information (PMI) for measuring the word association based on word frequency in a corpus. PMI is based on the notion of mutual information between two random variables $X$ and $Y$.

$$I(X, Y) = \sum_{x}\sum_{y} P(x, y) \log_2 \frac{P(x, y)}{P(x)P(y)} \tag{2.24}$$

The PMI (Fano and Hawkins, 1961) is a measure of how often two events $x$ and $y$ occur together assuming that they are independent. Church and Hanks (1990) applied this notion to measure the degree of statistical dependence between words by defining the PMI association between two words $w_i$ and $w_j$ in a given corpus.

$$PMI(w_i, w_j) = \log_2 \frac{P(w_i, w_j)}{P(w_i)P(w_j)} \tag{2.25}$$

where $P(w_i) = \frac{count(w_i)}{N}$ and $P(w_i, w_j) = \frac{count(w_i, w_j)}{N}$. $count(w_i)$ is the number of times word token $w_i$ appears in a given corpus and $N$ is the total number of word tokens in the corpus. $count(w_i, w_j)$ is the number of times that word $w_i$ and $w_j$ appear together in a given context window. In equation 2.25, the numerator tells how frequent two words appear together in the context, while the denominator specifies how frequent the two words appear independently. The ratio gives an estimate of how often two words tend to co-occur.

The similarity scores computed by PMI can be either positive or negative indicating that two words appear together more frequently or do not appear together. Negative PMI values measure the unrelatedness of words which is dependent on large corpora and usually unreliable (Church and Hanks, 1990). Thus, it is more common to use Positive Pointwise Mutual Information (PPMI) to measure the similarity between words by replacing all negative PMI values with zero (Dagan et al., 1994):

$$PPMI(w_i, w_j) = max(\log_2 \frac{P(w_i, w_j)}{P(w_i)P(w_j)}, 0) \tag{2.26}$$

Within the above basic definitions of PMI, other variations are proposed to count the appearance of words such as Normalized Google Distance (Gligorov et al., 2007), which relies on a search engine to count the occurrence of individual words and co-occurrence of words. Furthermore, word processing techniques such as word stemming, lemmatization, and spell checking are applied to identify the words in the corpus.

#### 2.2.4.2 Semantic Analysis

The PMI measures word association and encode the word similarity directly based on counting word frequency and word co-occurrence from a textual corpus. Other corpus-based similarity methods usually employ a Vector Space Model (VSM) (Turney et al., 2010) which represent words in vectors and word similarity are then computed based on vector similarity such as cosine similarity. The main idea is to derive the word vector representations from document collections based on word-document matrix. We introduce two main semantic analysis techniques Latent Semantic Analysis (LSA) (Landauer and Dumais, 1997) and Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch, 2007).

The main idea of ESA is representing words with high dimensional vectors constructed by explicit concepts. The original ESA (Gabrilovich and Markovitch, 2007) uses the encyclopedic knowledge base Wikipedia whose concepts are used to construct the vectors explicitly. Specifically, a word-document matrix $M$ is built and the matrix values are normalized with the TF-IDF weighting (Baeza-Yates et al., 1999). Since the documents represent Wikipedia documents and each document corresponds to a specific concept, the distribution of words in Wikipedia documents indicates the association between words and concepts. Thus the word-document matrix $M$ can be viewed as a word-concept matrix, while each row of the matrix represent a word vector. Then word similarity can be measured based on cosine similarity of word vectors. This word-concept matrix is usually sparseness, therefore, if two words have not appeared together in any Wikipedia document, their similarity value is zero.

LSA also measures word similarity based on how often two words appear in the same document but with dimension reduction. Moreover, LSA does not require the document collection representing meaningful concepts as the ESA. The document collection for LSA can be any form from normal text collection to a collection of sentences, paragraphs and word contexts. Given a document collection, LSA first builds a word-document matrix $M$ and then captures word correspondence from the matrix by operating dimension reduction, such as Singular Value Decomposition (SVD), on the word-document matrix. In consequence, after the dimension reduction, the semantic vector representation of words are obtained using latent topic vectors.

SVD is a common dimension reduction operation in linear algebra, which aims to find the correlations between rows and columns of the matrix. Formally, SVD factors the matrix $M$ into three matrices according to the following equation:

$$M_{|W| \times N} = U_{|W| \times K} \Sigma_{K \times K} V^T_{K \times N} \tag{2.27}$$

where $\Sigma_{K \times K}$ is the diagonal $K \times K$ matrix containing the $K$ singular values of $M$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$, and $U$ and $V$ are orthogonal matrices. The equation shows that the original matrix $M$ can be re-composed by multiplying the three matrices. $|W|$ and $N$ are the number of words and documents denoting the shape of the matrix $M$. Typically we can remove some insignificant dimensions by retaining only the $K^{'}$ largest singular values in $\Sigma$ and setting the remaining small ones to zero. The original $M$ is approximated by $K^{'}$ largest singular triplets and the new vector space becomes the latent semantic topical space. In consequence, the original word vector $v$ in matrix $M$ can be transformed into $K^{'}$ dimensional topic vectors through the following equation:

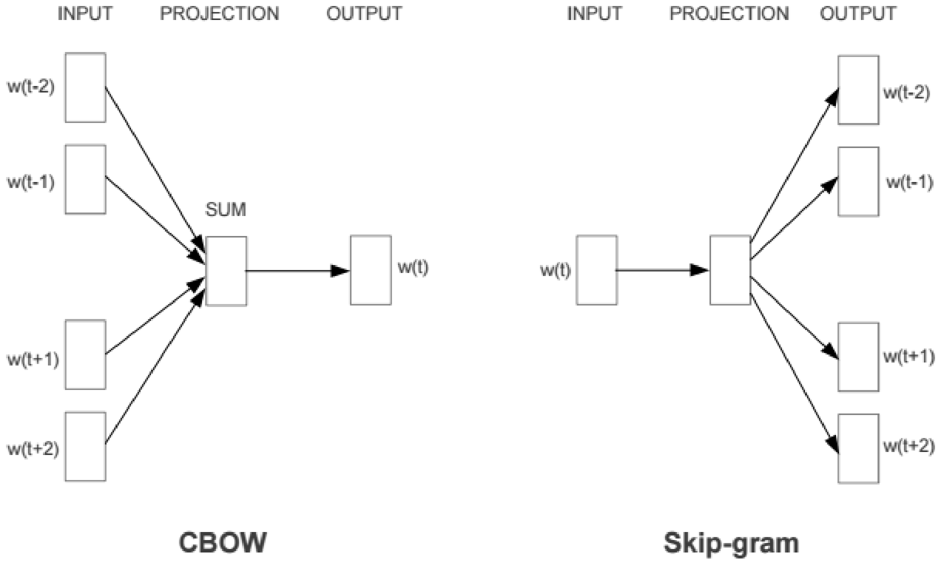
$$\hat{v} = v^T U_{|W| \times K^{'}} \Sigma^{-1}_{K^{'} \times K^{'}} \tag{2.28}$$

Within the lower dimensional topic vectors (e.g. $K^{'} = 300$), the word similarity can be computed using cosine similarity. In summary, LSA creates a VSM with latent topics and allows for a homogeneous representation of words. Because of this, LSA overcomes some of the drawbacks of the standard VSM for acIR such as sparseness and high dimension.

#### 2.2.4.3 Embedding Approaches

Word embedding refers to the methods that learn low-dimensional real-valued dense vectors of words, namely distributed representation (Williams and Hinton, 1986), aiming to represent words in real-valued continuous vector spaces and facilitate NLP tasks with effective generalized word features that take advantage from vector representations where similar words are close in the vector space. The traditional methods (e.g. LSA) first obtain a co-occurrence matrix and then perform dimension reduction. Recent embedding approaches

Figure 2.2: The CBOW architecture and Skip-gram Architecture of Word2Vec Embedding. Figure comes from Mikolov et al. (2013b)



adopt predictive methods that learn word vectors by predicting the contextual words of the target word. In this section, we introduce several state of the art word embedding models which are widely used in many NLP tasks.

Bengio et al. (2003) proposed word embedding based on feed forward neural network by predicting a word given precedent words, which is known as neural language model. The feed forward neural network takes words from a vocabulary and embeds them as vectors into a lower dimensional space. A recent consequent word embedding model Word2Vec (Mikolov et al., 2013b) simplifies the original model and significantly speed up the embedding training with efficient algorithms so that word embedding model can be trained with large corpora efficiently. Mikolov et al. (2013b) proposed two neural network training architectures for learning word embedding, Skip-gram and Continuous Bag of Words (CBOW), which are illustrated in Figure 2.2. The CBOW model trains word vectors in a neural network architecture which consists of an input layer, a projection layer, and an output layer to predict a word given its surrounding words within a certain context window size. The word embedding is yielded as a side-effect of the neural network training. Formally, assuming that we have a training corpus containing a sequence of $T$ training words $w_1, w_2, \dots, w_T$ and the corresponding vocabulary $V$, each word vector is trained to maximize the average log probability:

$$J_\theta = \frac{1}{T}\sum_{t=1}^{T} logp(w_t|w_{t-k}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+k}) \tag{2.29}$$

where $k$ is the context window size and $p(w_t|w_{t-k},\dots,w_{t+k})$ is the hierarchical softmax of the word vectors (Mikolov et al., 2013b). The skip-gram objective is to maximize the average log probability of the following function:

$$J_\theta = \frac{1}{T}\sum_{t=1}^{T}\sum_{-k\leq j\leq k,\neq 0} logp(w_{t+j}|w_t) \tag{2.30}$$

where $k$ is the size of the training window. The inner summation ranged from $-k$ to $k$ to calculate the log probability of the correctly predicting the word $w_{t+j}$ give the target word $w_t$. The outer summation covers all words in the training data. The prediction is performed through softmax which is illustrated as below:

$$p(w_{t+j}|w_t) = \frac{exp(v_{w_t}^T v'_{t+j})}{\sum_{w_i\in V} exp(v_{w_t}^T v'_{w_i})} \tag{2.31}$$

where $v_w$ denotes the input embedding vector and $v'_w$ denotes the output embedding vector. The softmax is trained using stochastic gradient descent and the gradient is obtained via back propagation (Rumelhart et al., 1988). Although the training process relies on a neural network based supervised prediction model, the real training results are the vector representation of words instead of the neural network prediction model. Because of such idea, word embedding is unsupervised and can be applied in various textual corpus without labeled dataset, which makes Word2Vec applicable to many NLP tasks. Furthermore, due to the simple neural network architecture and the use of hierarchical softmax, Word2Vec (Mikolov et al., 2013b) is able to address large dataset and the training is very efficient. As suggested by the Word2Vec (Mikolov et al., 2013b) authors, the CBOW model is more computationally efficient and suitable for larger datasets than the skip-gram model.

In addition, Doc2vec (Le and Mikolov, 2014) generalizes the Word2Vec (Mikolov et al., 2013b) to variable-length pieces of text, such as sentences, paragraphs and documents. This gives a collection of words with separate vector representations, which enables to memorize their semantic meaning. Both Skip-gram and CBOW are available in Word2Vec and Doc2Vec. Apart from the Word2Vec (Mikolov et al., 2013b), other embedding approaches such as GLOVE (Pennington et al., 2014) and PPMI-SVD (Levy and Goldberg, 2014) were proposed using matrix factorization (Pennington et al., 2014) and dimension reduction (Levy et al., 2015). Due to the increasing popularity and good performance reported in various applications, this thesis mainly use Word2Vec for training word embedding.

## 2.3 Summary

In this chapter, we have discussed the state of the art in domains related to development of the thesis. The goal is to make the reader familiar with the methodologies and technologies of selected domains and present the overview of those areas since they are used by the thesis to leverage the capabilities of KG and similarity. The review of specific elements of the state of the art related to disambiguation, classification and semantic search are presented individually later in each chapter.

In summary, firstly, we have shown the fundamental techniques of the Semantic Web in representing the Web of data. The overview of those techniques has illustrated the presence of semantic technologies that address modeling, publishing and querying web resources. The thesis builds on top of those achievements and proposes similarity metrics for measuring similarity between concepts in meta data level. For this reason, the Semantic Web is not the contribution area of the thesis but the technological background that is used as an enabler to extract semantic information and perform semantic query execution. Following the description of the Semantic Web, this chapter has presented an overview of the KG which is born under the hood of the Semantic Web. KG has used those enabling semantic technologies in the Semantic Web and community based collaborative knowledge base such as Wikipedia. The contribution of the thesis is based on the existing KGs, while those proposed contributions would facilitate the development of the KG-based applications. Thus, the formal definition of KG is given in order to provide enough background knowledge for presenting similarity framework and disambiguation framework in the later chapters. Finally, the semantic similarity techniques are detailed since they are the key focus of the thesis. The formal definition and classification of similarity methods are given so that the specific similarity domain is established. Then we present the state of the art similarity methods in two categories knowledge-based and corpus-based methods. The thesis contributes to the knowledge-based similarity by proposing new metric and semantic information with special focus on KG, while both type of similarity methods are considered in later application of disambiguation and classification.

# CHAPTER 3

# Semantic Similarity Framework

*Naturally, human tends to categorize things, events, location and people, by finding patterns they have in common. One of the intuitive way to relate two things is based on their similarity, which is a cognitive tool for human to understand the world of things. In computation, similarity is a metric that measures if one thing is similar to another, while word "similar" means having some common characteristics. In particular, semantic similarity is a special metric to quantify how much two objects are alike to each other respect to their meanings and taxonomical relation of object categorization.*

*Measuring semantic similarity between concepts has been proven to be beneficial to various of KG-based applications such as concept classification, QA, similarity-based search and recommendation. This chapter introduces a novel knowledge-based semantic similarity metric, called WPath, and a new semantic information in KG called graph-based IC. The combination of WPath metric and graph-based IC provides convenient semantic similarity computation of concepts in KG. The empirical evaluation is performed in the standard word similarity dataset, and the experimental results have shown that the improvement of WPath over other knowledge-based similarity metrics is statistical significant.*

## 3.1 Introduction

Some of the conventional semantic similarity metrics rely on measuring the similarity between concepts using hierarchical relations (Rada et al., 1989; Leacock and Chodorow, 1998; Wu and Palmer, 1994; Li et al., 2003). Semantic similarity between two concepts is then proportional to the length of the path connecting the two concepts. Path based similarity metric requires the structure of semantic network to generate a similarity score that quantifies the degree of similarity between two concepts. Concepts that are physically close to each other in a taxonomy are considered to be more similar than those concepts that are located far away. Some other semantic similarity metrics (Resnik, 1995; Jiang and Conrath, 1997; Lin, 1998) consider statistical IC of concepts computed from corpora in order to improve the performance of similarity metrics that are only based on the structure of concept taxonomy. IC is a measure of specificity of a concept. The higher values of IC are associated with more specific concepts (e.g., actor), while those lower values are more general (e.g., person). IC is computed based on frequency counts of concepts appearing in a textual corpus. Each occurrence of a more specific concept also implies the occurrence of the more general ancestor concepts.

In order to alleviate the weaknesses of both path based metrics and IC based metrics, we propose a novel semantic similarity method, namely WPath, combining the two methods. The main idea of the wpath semantic similarity method is to encode both the structure of the concept taxonomy and the statistical information of concepts. Furthermore, in order to adapt corpus-based IC methods to structured KGs, graph-based IC is proposed to compute IC based on the distribution of concepts over instances in KGs, and enable those semantic similarity metrics using IC to be used based on KGs without offline preparation of domain concept-annotated corpus. Consequently, using the graph-based IC in the WPath semantic similarity method can represent the specificity and hierarchical structure of the concepts in a KG. Within the graph-based IC, the WPath semantic similarity method can be used to compute semantic similarity between concepts in KGs only based on the structural knowledge of concepts and the statistical knowledge of instances in KGs.

In conclusion, this chapter considers the problem of measuring semantic similarity between concepts in KGs. The main contributions can be summarized as: (1) we propose a method for measuring the semantic similarity between concepts in KGs (Section 3.2); (2) we propose a method to compute IC based on the specificity of concepts in KGs (Section 3.3); (3) we evaluate the proposed methods in gold standard word similarity datasets (Section 3.4). Finally, we draw conclusions in Section 3.5.

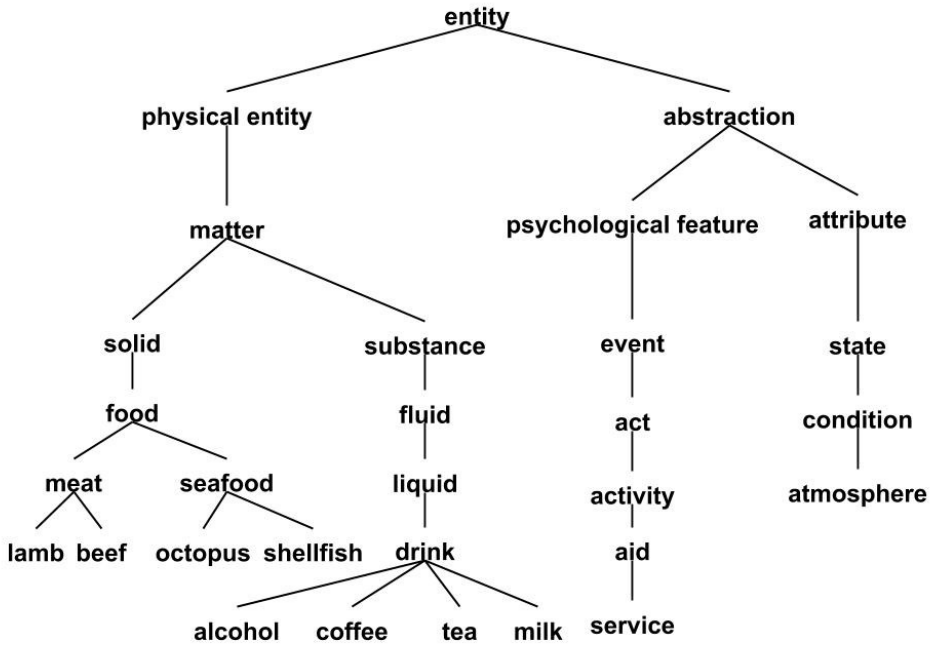


Figure 3.1: A Fragment of WordNet Concept Taxonomy

## 3.2 WPath Semantic Similarity Metric

Knowledge-based semantic similarity methods are mainly developed to quantify the degree to which two concepts are semantically similar using information drawn from concept taxonomies or ICs. Metrics take as input a pair of concepts, and return a numerical value indicating their semantic similarity score. Many applications rely on this similarity score to rank the similarity between different pairs of concepts. Take a fragment of WordNet concept taxonomy in Figure. 3.1 as an example, given the concept pairs of $(beef, lamb)$ and $(beef, octopus)$, the applications require similarity metrics to give higher similarity value to $sim(beef, lamb)$ than $sim(beef, octopus)$ because the concept *beef* and concept *lamb* are kinds of *meat* while the concept *octopus* is a kind of *seafood*. The semantic similarity scores of some concept pairs computed from the semantic similarity methods have been illustrated in Table. 3.1. It can be seen in this table how the row of concept pair $(beef, lamb)$ has higher similarity scores than the row of concept pair $(beef, octopus)$.

One of the drawbacks of conventional knowledge-based approaches (e.g. path or lch) in addressing such task is that the semantic similarity of any two concepts with the same path length is the same (uniform distance problem). As illustrated in Figure. 3.1 and Table. 3.1, based on the path and lch semantic similarity methods, $sim(meat, seafood)$ is the same as $sim(beef, lamb)$ and $sim(octopus, shellfish)$ because those concept pairs

| Concept Pairs | path | lch | wup | li | res | lin | jcn | wpath |
|---|---|---|---|---|---|---|---|---|
| beef - octopus | 0.200 | 2.028 | 0.714 | 0.442 | 6.109 | 0.484 | 0.071 | 0.494 |
| beef - lamb | 0.333 | 2.539 | 0.857 | 0.667 | 6.725 | 0.591 | 0.097 | 0.692 |
| meat - seafood | 0.333 | 2.539 | 0.833 | 0.659 | 6.109 | 0.760 | 0.205 | 0.662 |
| octopus - shellfish | 0.333 | 2.539 | 0.857 | 0.667 | 9.360 | 0.729 | 0.125 | 0.801 |
| beef - service | 0.071 | 0.999 | 0.133 | 0.000 | 0.000 | 0.000 | 0.050 | 0.071 |
| beef - atmosphere | 0.083 | 1.153 | 0.154 | 0.000 | 0.000 | 0.000 | 0.052 | 0.083 |
| beef - coffee | 0.111 | 1.440 | 0.429 | 0.168 | 3.337 | 0.319 | 0.066 | 0.208 |
| food - coffee | 0.143 | 1.692 | 0.500 | 0.251 | 3.337 | 0.411 | 0.095 | 0.260 |

Table 3.1: The Illustration of Semantic Similarity Methods on Some Concept Pair Examples

have equal shortest path length. Some knowledge-based approaches (e.g. wup or li) tried to solve the drawback by including depth information in concept taxonomy. Considering that the upper level concepts are more general than the lower level concepts in hierarchy, those approaches use the depth of concepts to give higher similarity value to those concept pairs which are located deeper in taxonomy. For example, the similarity of $(beef, lamb)$ is higher than the similarity of $(meat, seafood)$ based on semantic similarity method of wup and li, because the concept *lamb* and the concept *beef* are located deeper in the concept taxonomy (*lamb* and *beef* are sub-concepts of *meat*). Though using depth has been reported performance improvement compared to pure path length methods, for a given taxonomy such as the one in Figure. 3.1, many concepts share the same depth (hierarchical level) resulting in same similarity. For instance, as shown in Table. 3.1, based on the semantic similarity methods of wup and li, $sim(lamb, beef)$ is equal to $sim(octopus, shellfish)$ because of the same depth.

In order to solve the equal path length and depth problem, some knowledge-based approaches (e.g. res, lin, or jcn) proposed to include IC because different concepts usually have different IC values (e.g, the IC of *meat* is 6.725 and the IC of *food* is 6.109) so that the $sim(lamb, beef)$ is different from $sim(octopus, shellfish)$. Note that the IC in this section is based on corpus-based IC and its implementation details is described in Section 3.4.1. IC is a statistical method to measure the informativeness of concept. General concepts have lower informativeness thus have lower value of IC, while more specific concepts would have higher value of IC. For example, the IC of *meat* is higher than the IC of *food* because *meat* is a sub-concept of *food*. The idea of using IC to compute semantic similarity is that the more information two concepts share in common, the more similar they are. Using the IC of the LCS alone in the res method can represent the common information that two concepts share, however, the problem is that the similarity of any two concepts with the same LCS is the same. For example, based on res semantic similarity, although the concept pairs $(beef, lamb)$ and $(octopus, shellfish)$ have different similarity scores, the similarity scores of concept pairs $(meat, seafood)$ and $(beef, octopus)$, $(beef, coffee)$ and $(food, coffee)$ are the same because the LCS of the concept pairs are concept *food* and *matter*. Other methods (e.g., lin or jcn) tried to solve the drawback by including the IC of concepts being compared. However, only using the informativeness of concepts to represent the difference between concepts may lose the valuable distance information between concepts provided by the human experts who have created the concept taxonomy. It has been shown in our preliminary experiments that the path length between concepts in a taxonomy is a very effective feature in measuring semantic similarity of concepts. Furthermore, when the LCS of the concept pairs is the root concept *entity*, the li, res, and lin methods fail by generating 0 similarity score such as concept pairs $(beef, service)$ and $(beef, atmosphere)$. In addition, the lin and jcn methods are still missing the hierarchical level information. For instance, since the concept pairs $(meat, seafood)$ are more general than $(octopus, shellfish)$, the $(meat, seafood)$ is assumed to be less similar, however, the lin and jcn methods have given higher similarity score.

Considering both advantages and disadvantages of conventional knowledge-based semantic similarity methods, we propose a weighted path length (wpath) method to combine both path length and IC in measuring the semantic similarity between concepts. The IC of two concepts' LCS is used to weight their shortest path length so that those concept pairs having same path length can have different semantic similarity score if they have different LCS. The wpath semantic similarity method is illustrated in Eq.(3.1):

$$sim_{wpath}(c_i, c_j) = \frac{1}{1 + length(c_i, c_j) * k^{IC(c_{lcs})}} \tag{3.1}$$

where $k \in (0, 1]$ and $k = 1$ means that IC has no contribution in shortest path length. The parameter $k$ represents the contribution of the LCS's IC which indicates the common information shared by two concepts.

The proposed method aims to give different weights to the shortest path length between concepts based on their shared information, where the path length is viewed as difference and the common information is viewed as commonality. For identical concepts, their path length is 0 so their semantic similarity reaches the maximum similarity 1. As the path length between concepts in the concept taxonomy becomes bigger (bigger value of path length), the semantic similarity between concepts becomes smaller. The similarity score of the wpath is ranged in $(0, 1]$, which has improved the similarity score range in lch method and res method.

When the concepts have the same distance (equal path length), the more information two concepts share, the more similar they are. As shown in Table. 3.1, based on the wpath method, the similarity score of $(beef, lamb)$, $(meat, shellfish)$ and $(octopus, shellfish)$ are different based on their shared information, which shows the improvement of wpath over the path, lch, wup, and li methods. Although the wpath method is missing the depth, the LCS actually denotes the hierarchical level in taxonomy implicitly. Specifically, IC is a statistical method exploiting statistical occurrence information of concept, and the IC of LCS is similar to depth of concept indicating that the deeper level of concepts in the taxonomy are more specific, thus they are more similar. Moreover, concept's IC includes frequency of concepts so it has more various values than depth.

Since IC based metrics (e.g. res, lin and jcn) do not deal with the hierarchy of concepts, similarity scores computed by them lack of information about hierarchical levels and conceptual distance. As structural knowledge is retained in the wpath method, it is able to give higher similarity score to more specific concepts, but also give higher similarity score to those concepts sharing the same IC and located closer in taxonomy. In the example of $(beef, octopus)$, $(meat, seafood)$, since they share the same IC and $(meat, seafood)$ locates closer in the taxonomy, the wpath method has given higher similarity score to $(meat, seafood)$ than $(beef, octopus)$, which shows improvement of the wpath method over res method. The example of $(octopus, shellfish)$ and $(meat, seafood)$ shows that the wpath method has solved the hierarchical level problem of lin and jcn methods by giving higher similarity score to more specific concept pair when two concept pairs have the same path length.

In conclusion, the wpath semantic similarity method takes advantage of structure based methods (e.g. path, lch, wup and li) in representing the distance between concepts in a taxonomy, and overcomes the equal path and depth problem that would result in equal similarity scores for many concept pairs. By using the shared information (IC) between concepts to weight their path length, the wpath not only can retain the ability to show the distance between concepts based on a taxonomy, but also can acquire statistical information to tell the commonality between concepts when their conceptual structures in taxonomy are same. The $IC$ function in Eq.(3.1) denotes the general purpose IC which is used as weight for path length. According to different application scenarios, the $IC$ function can either be the corpus-based IC or the graph-based IC which will be introduced in the following section.

## 3.3 Graph-Based Information Content

Conventional corpus-based IC requires to prepare a domain corpus for the concept taxonomy and then to compute IC from the domain corpus in offline. The inconvenience lies in the high computational cost and difficulty of preparing a domain corpus. More specifically, in order to compute corpus-based IC, the concepts in the taxonomy need to be mapped to the words in the domain corpus. Then the appearance of concepts are counted and the IC values for concepts are generated. In this way, the additional domain corpus preparation and offline computation may prevent the application of those semantic similarity methods relying on the IC values (e.g., res, lin, jcn, and wpath) to KGs, especially when the domain corpus is insufficient or the KG is frequently updated. Since KGs already mined structural knowledge from textual corpus, we present a convenient graph-based IC computation method for computing the IC of concepts in a KG based on the instance distributions over the concept taxonomy. The graph-based IC is proposed to directly take advantage of KGs while retaining the idea of corpus-based IC representing the specificity of concepts. In consequence, the IC-based semantic similarity methods such as res, lin, jcn and the proposed wpath can compute the similarity score between concepts directly relying on the KG.

Concepts in KGs are usually represented as TBox and arranged into concept taxonomies. Those concepts categorize entity instances of ABox into different types via the special relation rdf:type. For example, the concept *movie* groups all movie instances in DBpedia. Moreover, if concept A is a parent concept of concept B and concept C in the taxonomy, then the set of instances of A is the union of the instances of B and C. In other words, a concept in KG can have multiple entity instances indicating the semantic type of those entities, while an instance can have multiple concepts to describe entity categories from general to specific. For instance, a DBpedia entity instance *dbr:Tom_ Cruise* can have several concepts describing its types from general to specific, *Person*, *Actor*, *AmericanFilmActor*. Intuitively, more general concepts occur more frequently in a KG such as concepts *organization* and *person*, while more specific concepts occur less frequently such as concepts *actor*, *university*, *scientist* and many others. Therefore, the proportion of the instances belonging to a specific concept in a KG can be used to measure the specificity of the concept for the given KG, which is similar to the idea of IC that measures the specificity (informativeness) of a concept over a corpus. Similar to the definition of conventional corpus-based IC, we extend the definition of IC in (Resnik, 1995) to KGs.

The graph-based IC in a KG is $IC_{graph}(c_i) = -logProb(c_i)$ where $Prob(c_i) = \frac{freq_{graph}(c_i)}{N}$. $N$ denotes the total number of entities in the KG. Let $entities(c_i)$ be the function to retrieve set of entities having type of $c_i$, the frequency of concept $c_i$ in theKG is defined as

$freq_{graph}(c_i) = count(entities(c_i))$ where $count(x)$ is a simple counting function measuring the cardinality of a set of entities.

The above definition of graph-based IC has defined the distribution of concepts over all the instances in KG. In particular, entity instances in KG are viewed as document collections and each instance denotes a document, while a collection of concepts describing each instance are viewed as terms in a document. Then the graph-based IC is computed as the frequency of those concepts over the document collections, whose idea is similar to the idea of Inverse Document Frequency (IDF) (Church and Gale, 1999) in IR, and the difference is the mathematical form. Both graph-based IC and IDF treat the less frequent concepts with higher importance. Since concepts in a taxonomy have hierarchical relations, the less frequent concepts specify more specific concepts.

Corpus-based IC methods may contain ambiguous meaning of concepts because it calculates IC by counting the occurrence of words over textual corpus, where words can be mapped to multiple concepts (ambiguous words). In comparison, graph-based IC contains specific meaning of concepts since KGs usually contain discriminated concepts to describe types of instances. Furthermore, similar to corpus-based IC, graph-based IC can be used in semantic similarity methods which need to employ ICs such as the res, lin, jcn and wpath similarity methods. If the LCS of two concepts appears less frequently in a KG, it means that two concepts are more similar. Using graph-based IC enables semantic similarity methods to compute semantic similarity between concepts only based on the specific KG without relying on additional corpora.

```
SELECT count(?e) as ?e WHERE {
          ?e rdf:type owl:Thing .
}

SELECT count(?e) as ?e WHERE {
          ?e rdf:type owl:Thing .
          ?e rdf:type <concept_uri> .
}
```

Table 3.2: Using SPARQL to count $N$ and $freq_{graph}(c_i)$.

Moreover, it is more convenient to compute graph-based IC than conventional IC. Since instances are linked to concepts through the rdf:type relation in a structured representation, it is convenient to retrieve all the entities in a KG belonging to a specific concept using structured query languages such as SPARQL[1]. This could be considered as online computation compared to the corpus-based IC that is required to compute in offline from textual corpus. Suppose that the SPARQL query language is implemented in the KG management system and the ontology classes are described using OWL[2], the total number of entities $N$ in the KG and the function $freq_{graph}(c_i)$ can be implemented using the SPARQL queries shown in Table 3.2.

The *concept_uri* denotes the URI link of the specific concept in the KG. Within the definition of graph-based IC and the SPARQL implementation of graph-based IC, it is convenient to compute the IC of a specific concept based on a KG. Note that the above SPARQL implementation is just an illustrative example, and the similar online computation of graph-based IC can be achieved by accessing a knowledge management system. In addition, apart from being used in semantic similarity methods, graph-based IC can also be used in other KG-based applications such as selecting the most specific type of a given entity. Furthermore, the definition of graph-based IC can be applied to conventional document analysis domain where the documents are tagged with hierarchical concepts such as the Open Directory Project[3], the Medical Subject Headings[4], the ACM Term Classification[5] and many others. This chapter focuses on applying graph-based IC in semantic similarity methods and leave its other applications as future work.

In summary, graph-based IC is proposed to be a possible substitution or complementary for the conventional corpus-based IC when the domain corpus is insufficient or online computation of IC is required. For those domains already containing annotated corpus such as Brown Corpus (Francis and Kucera, 1979) for WordNet, corpus-based IC could be used if it performs well in similarity metrics for domain applications. According to our experiments, graph-based IC is as effective as corpus-based IC although it is not outperforming, thus graph-based IC could be considered as a trade off of the efficiency, convenience and effectiveness, with corpus-based IC.

## 3.4 Semantic Similarity Evaluation

This section presents the evaluation of semantic similarity methods and graph-based IC in word similarity task. The goal of our experiments is to evaluate the proposed semantic

[1] https://www.w3.org/TR/sparql11-query/

[2] https://www.w3.org/TR/owl-ref/

[3] https://www.dmoz.org/

[4] https://www.nlm.nih.gov/mesh/

[5] https://www.acm.org/publications/class-2012

similarity method and graph-based IC in KGs. However, to best of our knowledge, currently there is no standard method and dataset to evaluate the performance of semantic similarity method and IC computation model for concepts in KG. Therefore, the commonly used word similarity datasets are used to evaluate the proposed semantic similarity method and graph-based IC based on WordNet and DBpedia. Moreover, the semantic similarity methods are evaluated in an aspect category classification task (Pontiki et al., 2015, 2016a) in order to evaluate their performance in a real application. This section presents the datasets, implementation, evaluation and provides a brief discussion about the obtained experimental results.

### 3.4.1 Datasets and Implementation

We collected several publicly available gold standard datasets for evaluating word semantic similarity, which are conventionally most commonly used and some recently most updated datasets. The description of collected datasets used in experiment are listed below.

- **R&G** (Rubenstein and Goodenough, 1965) is the first and most used dataset containing human assessment of word similarity. The dataset resulted from the experiment conducted in 1965 where a group of 51 students (all native English speakers) assessed the similarity of 65 pairs of words selected from ordinary English nouns. Those 51 subjects were requested to judge the *similarity of meaning* for two given words on a scale from 0.0 (completely dissimilar) to 4.0 (highly synonymous). It focused on semantic similarity and ignored any other possible semantic relationships between the words.
- **M&C** (Miller and Charles, 1991) replicated the **R&G** experiment again in 1991 by taking a subset of 30 noun pairs. The similarity between words was judged by 38 human subjects.
- **WS353** (Finkelstein et al., 2002) contains 353 pairs of words and 13 to 16 human subjects were asked to assign a numerical similarity score between 0.0 to 10.0 (0=totally unrelated and 10=very closely related). In fact, this dataset measures general relatedness rather than similarity because it considers other semantic relations (e.g. antonyms are considered as similar). We used this dataset because it has been perhaps the most commonly-used gold standard dataset for semantic similarity recently.
- **WS353-Sim** (Agirre et al., 2009) contains 203 pairs of words and is the subset of **WS353**. It has been identified by the authors to be suitable for evaluating semantic similarity specially.

- **SimLex** (Hill et al., 2014) is a recently released dataset consisting of 999 word pairs for evaluating semantic similarity specially. The dataset contains 111 adjective pairs (A), 666 noun pairs (N), and 222 verb pairs (V). We used 666 noun pairs in our experiment. Each pair of words was rated by at least 36 subjects (native English speakers) with similarity scores on scale from 0.0 (no similarity) to 10.0 (exactly mean same thing) and the average score was assigned as final human judgment score.

All the datasets described above contain a list of triples comprising two words and a similarity score denoting word similarity judged by human subjects. The human ratings on those word pairs have been proven to be highly replicable. The correlation obtained from M&C with respect to R&G's experiment was 0.97. (Resnik, 1995) replicated the M&C's experiment again in 1995, involving 10 computer science graduate students and post-doc researchers to assess similarity. The correlation with respect to the M&C's results was 0.96. This indicates that human assessment about semantic similarity between words is remarkably stable over a large time span and such datasets containing human ratings can be reliably used for evaluating semantic similarity methods. Since those datasets contain different coverage of word pairs, we use all the datasets for evaluation in order to present a more completed and objective experiment.

Those datasets are used for evaluating word similarity. However, the semantic similarity metrics presented in this paper are used for concepts, rather than words. We convert those concept-to-concept semantic similarity metrics into a word-to-word similarity metrics by taking the maximal similarity score over all the concepts which are the senses of the words (Resnik, 1995; Sánchez et al., 2012). This is based on the intuition that human subjects would pay more attention to word similarities (i.e., most related senses) rather than their differences while rating two non-disambiguated words (Sánchez et al., 2012), which has been demonstrated in psychological studies (Tversky, 1977). Polysemous words can be mapped to a set of concepts. Let $s(w)$ denote a set of concepts that are senses of word $w$, then the word similarity measure is defined as:

$$sim_{word}(w_i, w_j) = \max_{c_i \in s(w_i), c_j \in s(w_j)} sim_{concept}(c_i, c_j) \tag{3.2}$$

where $sim_{concept}$ can be any semantic similarity methods for concepts presented in this paper. This function is used to compute word similarity scores for each semantic similarity method to be evaluated in this section.

Moreover, we implemented all the knowledge-based semantic similarity methods and graph-based IC using WordNet version 3.0[6] and DBpedia 2014[7]. The semantic similarity

[6] https://wordnet.princeton.edu/

[7] http://dbpedia.org

methods li , jcn and the proposed wpath method are implemented based on WordNet NLTK interface[8]. We use the default implementation of other similarity methods in NLTK which are based on the perl module of WordNet::Similarity (Pedersen et al., 2004). We also use the NLTK's implementation of corpus-based IC using Brown Corpus (Francis and Kucera, 1979). For graph-based IC, we extracted 68423 WordNet concepts that have been used in YAGO (Hoffart et al., 2013) and used as DBpedia classes such as yago:Movie106613686. By computing the IC of those 68423 YAGO classes from DBpedia using the proposed graph-based IC, we can have the graph-based IC of those concepts in WordNet so that the graph-based IC can be evaluated using word similarity datasets. This graph-based IC computation is achieved by implementing a interface to compute IC using SPARQL queries which are executed in online SPARQL endpoint[9], including 4298433 entities. As a result, we developed a complete integrated framework to implement and evaluate semantic similarity methods for concepts in WordNet and DBpedia. All the implementations and resources, as well as the evaluation results, are published in Sematch[10] framework publicly.

### 3.4.2 Metrics and Evaluation

We follow the most established methodology for evaluating semantic similarity measures, which consists of measuring the Spearman correlation between similarity scores generated by the similarity methods and scores assessed by human. Note that both Spearman's and Pearson's correlations coefficients have been commonly used in the literatures. They are equivalent if rating scores are ordered and we use Spearman correlation coefficients in this paper for convenience. The conventional knowledge-based semantic similarity methods path (Rada et al., 1989) (Eq.(4.11)), lch (Leacock and Chodorow, 1998) (Eq.(2.6)), wup (Wu and Palmer, 1994) (Eq.(2.8)), li (Li et al., 2003) (Eq.(2.9)), res (Resnik, 1995) (Eq.(2.10)), lin (Lin, 1998) (Eq.(2.11)), jcn (Jiang and Conrath, 1997) (Eq.(2.13)) are used as compared methods and treated as baselines. A similarity measure is acknowledged to have better performance if it has higher correlation score (the closer to 1.0 the better) with human judgments, while it is acknowledged to be unrelated to human assessment if the correlation is 0. Since the Spearman's rank correlation coefficients produced by different semantic similarity methods are dependent on the human ratings for each dataset, we need to conduct statistical significance tests on two dependent (overlapping) correlations. We followed the Steiger's Z test (Steiger, 1980) used by Philipp et al. (Singer et al., 2013) to calculate statistical significance test between the dependent correlation coefficients produced

[8] http://www.nltk.org/

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

[10] https://github.com/gsi-upm/sematch/

by different semantic similarity methods, using a one-tailed hypothesis test for assessing the difference between two paired correlations. The cocor package of R[11] is used to calculate the statistical significance tests on dependent Spearman rank correlation coefficients. The statistical significance tests would determine whether the improvement in the correlation coefficient for each dataset is statistically significant.

In addition, the performance of IC is evaluated based on its performance of being used in semantic similarity methods. The IC computation method is acknowledged to be better if the semantic similarity method achieved better performance in using the IC. We compare the proposed graph-based IC to conventional corpus-based IC. The evaluation goal of graph-based IC is not to show that it outperforms the corpus-based IC, but rather to evaluate how graph-based IC can be exploited in IC-based semantic similarity metrics aiming to complement or substitute existing corpus-based IC methods in modern KGs.

In order to evaluate the wpath semantic similarity method and graph-based IC, word similarity datasets have been processed and split into Word-Noun, Word-Graph and Word-Type. For evaluating the wpath similarity metric, the Word-Noun task was created by mapping words in word similarity datasets to WordNet noun concepts. The performance of graph-based IC is compared to corpus-based IC based on their performance in the similarity metrics of wpath, res, lin and jcn. To compute graph-based IC, words in word similarity datasets need to be mapped to DBpedia concepts. However, many words are not used as concepts in DBpedia such as noon, madhouse or lad to name a few. In consequence, in order to compare wpath and res, the Word-Graph was created by mapping the LCS of word pairs to DBpedia concepts, while the Word-Type was created by mapping the word pairs to DBpedia concepts for comparing wpath, lin and jcn. The more detailed dataset split criteria are described as below:

- **Word-Noun**: Word pairs are chosen from all the original word similarity datasets if both words can be mapped to WordNet concepts, while unmapped word pairs are removed from the datasets. We run all the semantic similarity methods based on WordNet and corpus-based IC. This task evaluates the performance of semantic similarity methods.

- **Word-Graph**: Word pairs are further chosen from datasets if both words can be mapped to WordNet concepts and the LCS of mapped concepts is one of the extracted 68423 WordNet concepts which are used as DBpedia type. Apart from running all the semantic similarity methods based on corpus-based IC, we also use the graph-based IC

[11] https://cran.r-project.org/web/packages/cocor/index.html

| Task | R&G (65) | M&C (30) | WS353(353) | WS353-Sim(203) | SimLex(999) |
|---|---|---|---|---|---|
| Word-Noun | 65 | 30 | 348 | 201 | 666 |
| Word-Graph | 57 | 27 | 321 | 189 | 657 |
| Word-Type | 41 | 18 | 211 | 128 | 408 |

Table 3.3: Numbers of word pairs for Evaluation Tasks. The headline denotes the numbers of word pairs in original dataset

computed from DBpedia in the res method and the proposed wpath method. This task is able to evaluate the performance of the graph-based IC used in semantic similarity methods of res and wpath. This task is chosen because both res and wpath only rely on the IC of LCS.

- **Word-Type**: Word pairs are chosen if both words can be mapped to the extracted 68423 WordNet concepts used as entity type in DBpedia. We treat those mapped word pairs as DBpedia types. Then, all the semantic similarity methods are run using both corpus-based IC and graph-based IC. This task is able to evaluate the performance of graph-based IC used in semantic similarity of lin, jcn and wpath.

Table 3.3 shows the numbers of word pairs that are chosen from the original datasets in each task. In Word-Noun task, we generated word similarity scores of baselines and the proposed wpath (Eq.(3.1)) method using corpus-based IC. Furthermore, we experimented with different settings of $k$ in range of $(0, 1]$ with interval of 0.1. The Spearman correlations between the wpath method with different $k$ settings and human judgments are shown in Table 3.4. Each column denotes each dataset and each row denotes a specific $k$ value running the wpath method. Note that the bold values in each column denotes the highest correlation score for each dataset which is also same for the following tables. The Spearman correlations between baselines and human judgments are shown in Table 3.5. Each row represents a semantic similarity method and the columns denote the different datasets. The row wpath shows the highest correlation score from the Table 3.4 for each dataset. Note that the corpus in parentheses of each method denotes that the method has used corpus-based IC. The Word-Noun is a superset of Word-Graph and Word-Type, which contains complete word pairs and human ratings. In order to evaluate whether the proposed wpath semantic similarity method outperforms other semantic similarity methods, a statistical significance test based on Steiger's Z test (Steiger, 1980) has been carried out to analyse the results of Word-Noun task, using one tailed test and 0.05 statistical significance in each dataset. Table 3.6 shows the result of Steiger's Z significance test on the differences between Spearman correlations ($\rho$) of wpath method and other semantic similarity methods.

| wpath k | R&G(65) | M&C(30) | WS353(348) | WS353-Sim(201) | SimLex(666) |
|---|---|---|---|---|---|
| k = 0.1 | 0.747 | 0.703 | 0.279 | 0.538 | 0.486 |
| k = 0.2 | 0.746 | 0.696 | 0.326 | 0.621 | 0.497 |
| k = 0.3 | 0.776 | 0.737 | 0.345 | 0.640 | 0.550 |
| k = 0.4 | 0.785 | **0.740** | **0.349** | 0.647 | 0.573 |
| k = 0.5 | 0.790 | 0.738 | **0.349** | 0.649 | 0.482 |
| k = 0.6 | 0.789 | 0.732 | 0.348 | 0.648 | 0.589 |
| k = 0.7 | 0.791 | 0.723 | 0.348 | 0.650 | 0.596 |
| k = 0.8 | 0.794 | 0.728 | 0.344 | **0.652** | **0.603** |
| k = 0.9 | **0.795** | 0.726 | 0.335 | 0.644 | 0.601 |
| k = 1.0 | 0.781 | 0.724 | 0.314 | 0.618 | 0.584 |

Table 3.4: Spearman correlations with ground truth in Word-Noun Task for proposed wpath method with different settings of k.

| Method | R&G(65) | M&C(30) | WS353(348) | WS353-Sim(201) | SimLex(666) |
|---|---|---|---|---|---|
| path | 0.781 | 0.724 | 0.314 | 0.618 | 0.584 |
| lch | 0.781 | 0.724 | 0.314 | 0.618 | 0.584 |
| wup | 0.755 | 0.729 | 0.348 | 0.633 | 0.542 |
| li | 0.787 | 0.719 | 0.337 | 0.636 | 0.586 |
| res(corpus) | 0.776 | 0.733 | 0.347 | 0.637 | 0.535 |
| lin(corpus) | 0.784 | 0.752 | 0.310 | 0.609 | 0.582 |
| jcn(corpus) | 0.775 | **0.820** | 0.292 | 0.592 | 0.579 |
| wpath(corpus) | **0.795** | 0.740 | **0.349** | **0.652** | **0.603** |

Table 3.5: Word-Noun Task: Spearman correlations with ground truth of different semantic similarity methods.

In Word-Graph task, we computed the word similarity scores of baselines and the wpath method with the task setting of Word-Graph. Particularly, for the res and wpath we also

| | R&G(65) | | M&C(30) | | WS353(348) | | WS353-Sim(201) | | SimLex(666) | |
|---|---|---|---|---|---|---|---|---|---|---|
| Method | $\rho$ | p-value | $\rho$ | p-value | $\rho$ | p-value | $\rho$ | p-value | $\rho$ | p-value |
| path | .982 | .171 | .984 | .248 | .967 | .003 | .960 | .013 | .955 | .021 |
| lch | .982 | .171 | .984 | .248 | .967 | .003 | .960 | .013 | .955 | .021 |
| wup | .964 | .029 | .946 | .398 | .969 | .468 | .959 | .110 | .946 | .000 |
| li | .982 | .293 | .978 | .223 | .978 | .129 | .974 | .097 | .965 | .019 |
| res | .956 | .204 | .943 | .436 | .952 | .449 | .948 | .194 | .913 | .000 |
| lin | .956 | .314 | .969 | .353 | .903 | .040 | .900 | .038 | .944 | .021 |
| jcn | .876 | .296 | .890 | .067 | .831 | .026 | .845 | .023 | .916 | .029 |

Table 3.6: Steiger's Z significance test on the differences between Spearman correlations ($\rho$) of wpath method and other semantic similarity methods using 1-tailed test and 0.05 statistical significance .

| Method | R&G(57) | M&C(27) | WS353(321) | WS353-Sim(189) | SimLex(657) |
|---|---|---|---|---|---|
| path | 0.782 | 0.699 | 0.336 | 0.611 | 0.581 |
| lch | 0.782 | 0.699 | 0.336 | 0.611 | 0.581 |
| wup | 0.738 | 0.711 | 0.367 | 0.622 | 0.537 |
| li | 0.779 | 0.696 | 0.353 | 0.625 | 0.583 |
| lin(corpus) | 0.776 | 0.736 | 0.324 | 0.596 | 0.578 |
| jcn(corpus) | 0.762 | **0.794** | 0.308 | 0.589 | 0.576 |
| res(corpus) | 0.765 | 0.713 | 0.365 | 0.626 | 0.530 |
| res(graph) | 0.721 | 0.717 | 0.315 | 0.543 | 0.373 |
| wpath(corpus) | **0.796** | 0.714 | **0.370** | **0.647** | **0.600** |
| wpath(graph) | 0.788 | 0.776 | 0.336 | 0.620 | 0.581 |

Table 3.7: Word-Graph Task: Spearman correlations with ground truth of different semantic similarity methods.

used graph-based IC, while the lin and jcn only used corpus-based IC. The Spearman correlations between similarity methods and human judgements for Word-Graph task are shown

| Method | R&G(41) | M&C(18) | WS353(211) | WS353-Sim(128) | SimLex(408) |
|---|---|---|---|---|---|
| path | 0.679 | 0.621 | 0.353 | 0.601 | 0.616 |
| lch | 0.679 | 0.621 | 0.353 | 0.601 | 0.616 |
| wup | 0.613 | 0.606 | 0.357 | 0.589 | 0.538 |
| li | 0.673 | 0.614 | 0.361 | **0.612** | 0.612 |
| res(corpus) | 0.667 | 0.679 | 0.355 | 0.595 | 0.540 |
| res(graph) | 0.674 | 0.704 | 0.294 | 0.487 | 0.381 |
| lin(corpus) | 0.642 | 0.696 | 0.322 | 0.539 | 0.592 |
| lin(graph) | 0.624 | 0.661 | 0.305 | 0.517 | 0.534 |
| jcn(corpus) | 0.676 | **0.805** | 0.342 | 0.546 | 0.594 |
| jcn(graph) | 0.309 | 0.324 | 0.241 | 0.440 | 0.331 |
| wpath(corpus) | 0.691 | 0.669 | **0.367** | 0.606 | **0.625** |
| wpath(graph) | **0.717** | 0.765 | 0.353 | 0.601 | 0.616 |

Table 3.8: Word-Type Task: Spearman correlations with ground truth of different semantic similarity methods.

| Setting | R&G | M&C | WS353 | WS353-Sim | SimLex |
|---|---|---|---|---|---|
| Word-Graph IC-Corpus | k=0.9 | k=0.5 | k=0.7 | k=0.8 | k=0.8 |
| Word-Graph IC-Graph | k=0.9 | k=0.5 | k=0.8 | k=0.9 | k=1.0 |
| Word-Type IC-Corpus | k=0.8 | k=0.5 | k=0.7 | k=0.9 | k=0.9 |
| Word-Type IC-Graph | k=0.6 | k=0.6 | k=1.0 | k=1.0 | k=1.0 |

Table 3.9: Spearman correlations with ground truth in Word-Noun Task for proposed wpath method with different settings of k.

in Table 3.7. The graph in parentheses of methods denote that the method has used the graph-based IC. In Word-Type task, we computed the word similarity scores of baselines and the wpath method with the task setting of Word-Type. Apart from corpus-based IC, we also used graph-based IC for the methods of res, lin, jcn, and wpath. The difference between the methods res, wpath and methods of lin, jcn is that the previous two only use the IC of LCS while the latter two also use the IC of individual concepts. The Spearman correlations between similarity methods and human judgments for the Word-Type task are shown in Table 3.8. We also experimented with different k settings of wpath method in the Word-Graph task and Word-Type task for both corpus-based IC and graph-based IC. In Table 3.7 and Table 3.8 we reported the best results of wpath method for each task and each dataset, while Table 3.9 shows the specific settings of wpath method achieved best result in each task and each dataset. Within the evaluation of three tasks and corresponding results, we then analyse the results in the following section.

### 3.4.3 Result Analysis and Discussion

Our main hypothesis in the experiments is that the proposed semantic similarity method wpath will improve over the baselines and show high correlation to human assessments. The second hypothesis is that the proposed graph-based IC computation method is effective compared to the conventional corpus-based IC, which means the graph-based IC needs to show close performance or outperforming in some cases.

Table 3.5, 3.7 and 3.8 show that all the semantic similarity methods have high correlation with human judgements and the proposed wpath semantic similarity method outperforms the baselines in most of cases except the M&C dataset and WS353-Sim dataset in WordType task. Moreover, from the three tables we observed that the **jcn** method performed exceptionally best in the M&C dataset, however in other datasets it performed not as good as the one in M&C dataset. It is probably because of the small word pair sample in M&C dataset. It was also surprising that the **li** method had performed best only in WS353-Sim in Word-Type task. It may be caused by the specific subset of the dataset.

In Table 3.6, on R&G dataset, the significance test shows the improvement of wpath over wup (the p-value of each test is below the significance level of 0.05), while indicates no statistical significance differences with other methods. Regarding the M&C dataset, although the jcn method performs best, the result of statistical significance test indicates that no statistic significant differences between wpath and jcn (p-value > 0.05). On WS253-Sim and WS353 datasets, it is clear that the wpath has statistical significant improvement over the path, lch, jcn and lin. Finally, on SimLex dataset, the wpath has statistical significant improvement over all other semantic similarity metrics. In general, from the results of our experiments, we observed that different semantic similarity metrics have performed differently in different datasets. The wpath similarity metric has obtained the best performance in 4 out of 5 datasets (ranked as second in M&C only containing 30 word pairs). This shows that the wpath similarity metric has provided a stable performance in all datasets. Considering that SimLex is the largest dataset for semantic similarity, bigger than the combination of all other datasets, we may conclude that the wpath method has produced statistically significant improvement over other semantic similarity metrics.

From Eq.(3.1) we know when $k = 1$ the proposed wpath method is equivalent to path method. As the value of $k$ becoming smaller, IC starts to have bigger influence. Even with low or high values of $k$, $k$ contributes to solve the uniform distance problem of the path method illustrated in Table. 3.1. It has been shown in Table 3.4 and Table 3.9 that the best $k$ has smaller value in R&G and M&C datasets. As pure IC-based semantic similarity methods also achieved better performance in those two datasets, probably the human ratings of word pairs in those datasets care more on IC or general relevance. Based on this observation, the parameter $k$ actually defines for a given KG the balance among hierarchical structure and statistical information for calculating semantic similarity. Its values can provide insight about which metrics perform better in a given KG. For high values of $k$, structural metrics will provide a better result and for low values of $k$, IC metrics perform better.

Different KGs have different concept taxonomies and different distributions of instances over concepts. Even in a given concept taxonomy, concepts are not equally structured, such as various density of sub-concepts and different hierarchical levels of concepts. This can be shown in Fig. 3.1. Given that applications usually use a group of concepts from a taxonomy (e.g. restaurant domain), the specific value of $k$ should be selected for a specific domain (e.g. a subgraph of a KG) that reflects the concept structure and IC of that domain. In consequence, the selection of $k$ would be the optimization of $k$ for a specific group of concepts. For those concepts having human ratings, $k$ can be adjusted empirically or learned automatically by comparing to human ratings. For those concepts without human ratings, $k$ should be determined according to the specific domain application, in which $k$ can be selected empirically or learned automatically based on application performances.

Regarding to the graph-based IC, we observed that it performed better in Word-Type task than Word-Graph task. It is also shown in Table 3.7 and 3.8 that the graph-based IC has better performance in res, lin and wpath methods than jcn. It is shown in Table 3.8 that graph-based IC may not be suitable for the **jcn** method, and the graph-based IC achieved the best performance in R&G and M&C dataset in Word-Type task while had a similar result in other datasets compared to corpus-based IC. Consequently, we may conclude that the graph-based IC computation method is effective compared to conventional corpus-based IC in measuring word similarity but not always outperforming. Moreover, graph-based IC has a number of benefits, since it does not requires a corpus and enables online computing based on available KGs. Besides, graph-based IC metrics can benefit from the success of open linked data, and the continuous growth of available KGs.

## 3.5 Summary

According to the evaluation experiments in word similarity datasets, compared with the previous state of the art semantic similarity methods, the wpath method results in statistical significant improvement of correlation between computed similarity scores and human judgements. The proposed graph-based IC has shown to be effective as the corpus-based IC so that it could be used as the substitution of the corpus-based IC in KGs. Measuring semantic similarity of concepts is a crucial component in many applications which has been presented in the introduction. In this chapter, we present wpath semantic similarity method combining path length with IC. The basic idea is to use the path length between concepts to represent their difference, while to use IC to consider the commonality between concepts. The experimental results show that the wpath method has produced statistically significant improvement over other semantic similarity methods. Furthermore, graph-based IC is proposed to compute IC based on the distributions of concepts over instances. It has been shown in experimental results that the graph-based IC is effective for the res, lin and wpath methods and has similar performance as the conventional corpus-based IC. Moreover, graph-based IC has a number of benefits, since it does not requires a corpus and enables online computing based on available KGs.

CHAPTER 4

# Semantic Disambiguation Framework

*KGs are labeled graphs containing tremendous string names for concepts and named entities. Concepts represent basic unit of meanings and denote abstraction of entities. Since ambiguous words can be mapped to multiple concepts and many named entities in KGs share same names, semantic disambiguation of words and named entities plays an important role for developing KG-based applications that require to link natural language texts to concepts and entities in KG.*

*In this chapter, we investigate disambiguation methods respectively for words and named entities. As WordNet concepts have been integrated into many KGs, we propose a novel embedding approach to represent WordNet concepts and words in a shared vector space in order to discriminate ambiguous words to correct concepts based on context similarity. Regarding named entity disambiguation, we exploit various semantic similarity methods for entity disambiguation based on entity descriptions and categories.*

## 4.1 Word Sense Disambiguation

Word embeddings have recently become popular in a number of NLP tasks, including sense disambiguation. Sense embedding techniques learn a distributed vector for each sense of a word. Existing research has used WordNet mainly as a sense inventory to initialize the vector representation of synsets that are later augmented with annotated corpus. In this section, we propose a different approach that transforms word embeddings to the synset level and leverages the knowledge of WordNet and sense-annotated datasets by creating enriched synset profiles based on these semantic networks, including their semantic relationships and examples. The approach has been validated with WSD datasets reporting its effectiveness in representing synsets and words in the shared semantic vector space.

Recent advances of unsupervised word embedding techniques (Collobert and Weston, 2008; Mikolov et al., 2013b; Pennington et al., 2014) have proven to be beneficial to various NLP applications such as word similarity (Mikolov et al., 2013b), machine translation (Zou et al., 2013), syntactic parsing (Weiss et al., 2015), and question answering (Bordes et al., 2012) to name a few. In general, word embedding is actually a new branch of corpus-based distributional semantics models (Turney et al., 2010) and provides a distributed semantic representation of words by capturing both semantic and syntactic information of words (e.g. context and collocations) from large textual corpora using neural language model (Bengio et al., 2003). In most word embedding applications, words are embedded into lower dimensional dense vector space and each word corresponds to a single vector without considering the word polysemy. This limitation of word vector representation is a main hamper to apply word embedding for applications that require to discriminate various meanings of the same word, such as WSD (Navigli, 2009; Iacobacci et al., 2016).

To distinguish different word meanings, conventional unsupervised knowledge based WSD systems discriminate different synsets of polysemous words based on semantic similarity metrics between synsets (Navigli, 2009). Those metrics compute similarity scores based on the taxonomical structure of WordNet, and statistical information contents computed from textual corpora (Zhu and Iglesias, 2017). These WSD systems are not dependent on general sense or word representation, but only rely on similarity metrics and underlying semantic resources. Recent works have tried to learn distinct representations for individual word senses based on clustering approaches (Reisinger and Mooney, 2010; Huang et al., 2012), mixed or hybrid sense and word embedding (Chen et al., 2014; Iacobacci et al., 2015), new embedding architectures (Mancini et al., 2016) and various knowledge sources (Rothe and Schütze, 2015). Embedding-based sense and word representation models have shown promising performance in WSD tasks (Iacobacci et al., 2016), and they have better generalization power of the vector representation of words, compared to similarity-based approaches. However, clustering approaches suffer from determining proper number of clusters and corresponding to sense inventories, while existing embedding approaches rely on external word embedding and treat word and synset uniformly without considering semantic relationships of synsets.

This chapter proposes a novel approach, called the Synset2Vec model, that aims at leveraging some of the drawbacks previously identified and exploit the existing knowledge in WordNet and sense annotated datasets. Our approach is inspired in the clustering approaches that use contextual words for representing word senses. We aim at providing a joint embedding representation of both synsets and contextual words. To this end, we propose to create enriched synset profiles based on semantic networks such as WordNet and sense annotated datasets, including their semantic relationships and examples. Based on these enriched profiles, we apply a joint embedding approach to learn synsets and contextual words. The proposed vector representation benefits from its application in various applications.

Our contributions are threefold: (1) we propose a uniform embedding representation for different semantic networks such as WordNet (Miller, 1995) (e.g. synsets, glosses, usage examples, lemmas, ontological information) and sense-annotated corpus such as SemCor (Miller et al., 1993). Each synset has a specific vector representation and contextual words are shared among semantically related synsets; (2) we propose an algorithm to obtain enriched synset profiles from available datasets; (3) we validate the approach in a fine-grained WSD task (Alessandro Raganato and Navigli, 2017) showing its effectiveness to capture the semantic similarity between synset-synset, synset-word and synset-text.

### 4.1.1 Related Works

Distributed word representations (Williams and Hinton, 1986) aim to represent words in real-valued continuous vector spaces and facilitate NLP tasks with effective generalized word features that take advantage from vector representations where similar words are close in the vector space.

Compared to conventional distributional semantic models, such as LSA (Landauer and Dumais, 1997) and Latent Dirichlet Allocation (Blei et al., 2003), Bengio et al. (2003) proposed word embedding based on feed forward neural network by predicting a word given precedent words, which is known as neural language model. A recent consequent word embedding model Word2Vec (Mikolov et al., 2013b) simplifies the original model and significantly speed up the embedding training with efficient algorithms so that word embedding model can be trained with large corpora efficiently. Doc2vec (Le and Mikolov, 2014) generalizes the Word2Vec to variable-length pieces of text, such as sentences, paragraphs and documents. This gives a collection of words with separate vector representations, which allows to memorize their semantic meaning. Mikolov et al. (2013b) proposed two neural network models, Skip-gram and CBOW, which are both available in Word2Vec and Doc2Vec. The CBOW model combines the representations of surrounding words to predict the target word, while the Skipe-gram model uses the target word to predict the surrounding words. The Synset2Vec is built based on Doc2Vec and uses the CBOW model.

In order to overcome the limitation of one vector representation per word and consider the distinct meanings of polysemous words, numerous efforts have been made to represent more fine-grained word senses. (Reisinger and Mooney, 2010) proposed a clustering-based method that assigns multiple vectors to polysemous words representing different word senses. Following this idea, (Huang et al., 2012) employed probabilistic neural language models to learn distributed vectors for semantic word representation. These methods rely on word context cluster to determine the word sense which suffers from determining proper number of clusters and corresponding to proper word senses in sense inventory for WSD directly. The more straightforward way to learn word sense representations is to train semantic representation of word senses with a publicly available sense inventory such as WordNet (Miller, 1995). (Chen et al., 2014) leveraged existing word embedding of Word2Vec trained from large text corpora. They used the definitions of synsets in WordNet to initialize the vector representation of synsets by averaging word vectors. Then they updated the synset vectors with Skip-gram training model, where word vectors are replaced with synset vectors if certain disambiguation confidence is satisfied. Similarly, AutoExtend (Rothe and Schütze, 2015) trains mixed embedding of words, lexemes, and synsets with two steps: a first training of Word2Vec embedding and a second training of lexemes and synsets. SensEmbed (Iacobacci et al., 2015) uses BabelNet (Navigli and Ponzetto, 2012) as sense catalog and relies on CBOW model to learn sense embeddings from automatically sense-annotated corpora, where senses are discriminated with a WSD system, Babelfy (Moro et al., 2014). A consequent SW2V model (Mancini et al., 2016) extends the input and output layer of the neural network architecture with word senses in CBOW model, so that word and sense embeddings can be trained jointly exploiting knowledge from both text corpora and large semantic networks. Apart from representing senses and words in the same semantic vector space, Nasari (Camacho-Collados et al., 2016) presented joint representation of senses, words and named entities. Moreover, word embeddings is also studied to be combined with supervised WSD systems (Iacobacci et al., 2016) and semi-supervised WSD systems (Yuan et al., 2016).

In contrast to the current semantic vector representations of senses and words, Synset2Vec exploits WordNet data thoroughly without relying on external word embeddings. Also, Synset2Vec uniforms WordNet data and its sense-annotated data into the same form of synset profiles so that embedding training only need to run once with the same neural network structure. Furthermore, synset expansion is included using ontological knowledge. The expansion can be performed in both WordNet data and sense-annotated data, so that more training data are available by creating more associations between synsets and words. Synset2Vec aims to create a common vector space of synsets and words following the clustering idea for WSD. With such consideration, it groups definitions, examples, annotated sentences together and corresponds them to related synsets through synset profiles. In this way, Doc2Vec CBOW model is used to train synset and word embedding jointly from profile documents. Thus, word senses can be discriminated using the vector distance between synsets and contextual words.

### 4.1.2 The Synset2Vec Embedding

Synset2Vec is composed of two processes. First, the embedding data is prepared by creating enriched synset profiles that exploit various information sources, from both sense inventory and sense-annotated corpus. Second, the joint embedding model of synsets and words is trained using the prepared embedding data.

#### 4.1.2.1 Embedding Data

The embedding data $D$ consists of $N$ synset profiles $D = \{P_1, P_2, \ldots, P_N\}$. Each synset profile $P_k = \langle S_k, W_k \rangle$ is composed by a tuple containing a set of synsets $S_k$ and a sequence of word tokens $W_k$. Both sense inventory and sense-annotated corpora are transformed in order to construct the uniform embedding data $D$. The synset profile creation method is illustrated in Algorithm 1. The details are introduced in this section.

The lexical database WordNet (Miller, 1995) is used as sense inventory. Each sense entry is denoted as a synset representing one specific meaning and consists of various textual information sources, such as a set of lemma words (synonyms), synset gloss and synset usage examples. Each synset in WordNet results in a synset profile $P_k$, while $S_k$ is composed by the synset and the $W_k$ is created from extracting textual data from synset's textual data. Moreover, as WordNet can be conceptualized as a ontology containing semantic relations such as hypernym and hyponymy (is-a relation), meronym and holonym (part-of relation), we further expand $S_k$ by including related synsets. This is represented by function *expand* in Algorithm 1. We assume that a synset's super-concepts and related concepts (holonyms

**Algorithm 1** Synset Profile Preparation

```
procedure PREPARE($WordNet, Corpus$)
    $D \leftarrow \emptyset$
    for all $s \in WordNet$ do
        $W_i \leftarrow words(s.text)$
        EXPAND($s, S_i$)
        $D \leftarrow \langle S_i, W_i \rangle$
    end for
    for all $sent \in Corpus$ do
        $W_i \leftarrow words(sent)$
        for all $s \in sent.annotations$ do
            EXPAND($s, S_i$)
        end for
        $D \leftarrow \langle S_i, W_i \rangle$
    end for
    return $D$
end procedure
procedure EXPAND($synset, \Sigma$)
    $\Sigma \leftarrow synset$
    $\Sigma \leftarrow synset.holonyms$
    $\Sigma \leftarrow synset.meronyms$
    for all $s \in hypernyms(synset)$ do
        EXPAND($s, \Sigma$)
    end for
end procedure
```

and meronym) also have semantic association with the textual words of the synset. An example of synset profile of *car.n.01* is shown in Table 4.1.

In addition to the WordNet data, the sense-annotated data such as SemCor (Miller et al., 1993) is used for populating the synset profiles. Each annotated sentence in the corpus is used for generating a synset profile. For example, the first annotated sentence in SemCor is "the group state Friday an probe of Atlanta's late primary produce", with its corresponding annotations being *group.n.01, state.v.01, friday.n.01, probe.n.01, atlanta.n.01, late.s.03, primary.n.01, produce.v.04*. The original text of sentences is used to compose $W$ by extracting informative words including annotated words and non-stopwords. The annotated synsets construct the $S$ by performing synset expansion.

| | |
|---|---|
| **Words** $W$ | car, auto, automobile, machine, motorcar, motor, vehicle, wheel, propel, internal, combustion, engine, car, get, work. |
| **Synsets** $S$ | car.n.01, motor_vehicle.n.01, hood.n.01, window.n.02, accelerator.n.01, car_mirror.n.01, air_bag.n.01, ...etc. |

Table 4.1: An example of automatically constructed synset profile of the synset *car.n.01* from WordNet.

In summary, each profile $P_k$ in $D$ denotes a semantic association between synsets and words, where $W_k$ is a explicit word group that represents the meaning of each synset in $S_k$.

#### 4.1.2.2 Embedding Model

Once the data $D$ is obtained, we then apply an embedding approach to train synset and word embedding jointly so that the associations between synsets and words are represented as their inter-connectivity in the shared vector space.

Formally, given a training profile having a set of synsets $S = \{s_1, s_2, \ldots s_j\}$, and a sequence of word tokens $W = \{w_{t-k}, \ldots, w_t, \ldots, w_{t+k}\}, -k \leq i \leq k, i \neq 0$, where $w_t$ is the target word, the training goal is to maximize the average log probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-k\leq i\leq k, i\neq 0} logp(w_t|S^t, W^t) \tag{4.1}$$

where $k$ is the context window size and $T$ is the size of the training words. $p(w_t|S^t, W^t)$ is the hierarchical softmax of the synset vectors and word vectors, which are trained using stochastic gradient descent and the gradient is obtained via back propagation (Rumelhart et al., 1988). As shown in Figure 4.1, every synset and word are mapped to unique vectors and their average or concatenation are used to predict the target word in a context. Since the main training goal is to model the synset-word association in the vector space, synset vectors contribute to the prediction task of the target word together with contextual words sampled from synset profiles. Synsets are viewed as special tokens recording the semantic topics of target words as well as contextual words. The contextual words for prediction are fixed-length and sampled from a sliding window over the profile. The word vectors are shared across all the synset profiles as long as the word occurs in the profile, while the synset vectors are shared across those profiles containing the same synset.

In Synset2Vec, because synsets are viewed as special words for recording the semantic

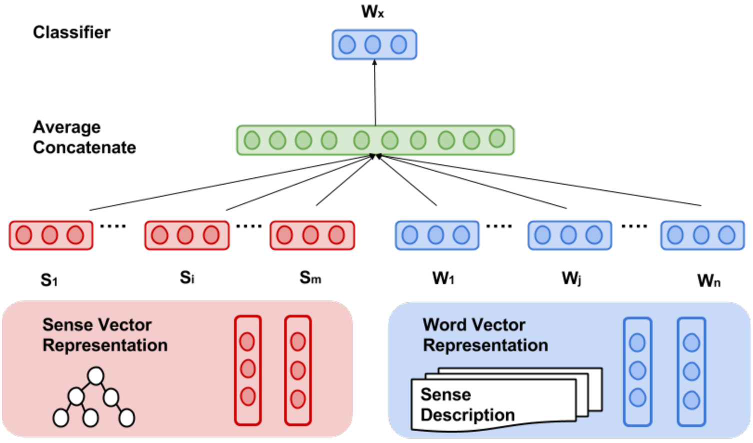


Figure 4.1: Jointly Learning Embeddings of Senses and Words in WordNet

topics of a group of words (words of the profile), synsets can be viewed as explicit topics given to those word clusters. Moreover, as we use synset expansion for generating embedding data, more general synsets (super-concepts) will appear more frequently in profiles, so they will be trained together with larger numbers of words, while more specific synsets will be trained less frequently. As a result, in the trained semantic vector space, general synsets are mapped to the locations that are closer to larger group of words while specific synsets are located closer to more specific words. In other words, general synsets can be viewed as representation of larger word clusters and specific synsets correspond to smaller word clusters, which are a subset of larger word clusters. With such property, Synset2Vec model can be used to compare various semantic distance of synset-synset, synset-word, and synset-text (text vector is normalized as the average of word vectors).

In addition, the Synset2Vec model is built on top of Doc2Vec (Le and Mikolov, 2014), which treats variable length sized text (e.g. documents, paragraphs and sentences) as special tokens and it is trained with normal words jointly on top of Word2Vec (Mikolov et al., 2013b). In comparison, synsets are similar to those documents, paragraphs, and sentences, acting as memory for recording the main topics of co-training words. Furthermore, training of joint vector representations of synsets and words are actually achieved using the co-occurrence statistics of synsets and words in the profiles, which is based on distributional semantics hypothesis (Turney et al., 2010). A synset and a word are assumed to be more similar if they appear together more frequently. Correspondingly, in the shared vector space, the vector distance of frequently collocated synsets and words are smaller.

### 4.1.3 Fine-Grained Sense Disambiguation

WSD is a core task in Natural Language Understanding which aims to automatically assign the most appropriate word sense (a specific word meaning) to a polysemous target word in a given context. A correct sense is usually modeled as a proper entry in a given sense inventory such as WordNet. There are many WSD systems including supervised systems (Zhong and Ng, 2010; Iacobacci et al., 2016; Melamud et al., 2016) and knowledge-based systems (Lesk, 1986; Navigli, 2009; Pilehvar and Navigli, 2014). As we want to evaluate the joint vector representation of synsets and words, we have implemented four unsupervised knowledge-based WSD frameworks using contextual similarity (Navigli, 2009) and graph-based ranking (Sinha and Mihalcea, 2007). These approaches exclusively depend on the different vector similarities of synsets, words and texts.

The implementation of context-based WSD is based on a knowledge-based approach using semantic similarity metrics (Navigli, 2009). Following, we define the context-based WSD framework formally. Given that $Sim$ is a general similarity function measuring the similarity between a candidate synset and contextual text, let $Senses(w_i)$ denote all the candidate synsets of a target word $w_i$, let $T = (w_1, \ldots, w_n)$ be the contextual text, and let $\hat{S}$ be the correct synset of $w_i$ that is determined by maximizing the following similarity function:

$$\hat{S} = \underset{s_i \in Senses(w_i)}{argmax} \; Sim(s_i, T) \tag{4.2}$$

According to the different similarity metrics namely synset-synset, synset-word and synset-text; we develop the WSD1, WSD2 and WSD3 systems which respectively implement the $Sim$ functions.

**WSD1** implements the $Sim$ function using synset-synset similarity metric, based on cosine similarity of synset vectors, expressed in the following expression:

$$Sim(s_i, T) = \frac{1}{|T|} \sum_{w_j \in T} \max_{s_k \in Senses(w_j)} cosine(s_i, s_k) \tag{4.3}$$

WSD1 also considers the polysemy of contextual words, selecting the meaning of contextual words by computing the maximum of their synset-synset similarity with the target word. Consequently, the context similarity $Sim$ is achieved by first choosing the closest sense of a word in context, and averaging the synset-synset vector similarity scores of contextual words. Note that we only select those contextual words having similarity scores higher than a threshold 0.2. This aims to remove the noise introduced by irrelevant words, such as stopwords. WSD1 system aims to evaluate the synset vector representation by investigating the inter-connectivity between synsets in the vector space.

**WSD2** implements the *Sim* function using synset-word similarity based on cosine similarity of synset vector and word vector. In this model, synsets are acting like topics that are embedded into specific group of words. Through the measure of the vector similarity between synsets and words we can illustrate the inter-connectivity between synsets and words in a shared semantic vector space. Following this idea, the corresponding *Sim* function is shown as below:

$$Sim(s_i, T) = \frac{1}{|T|} \sum_{w_j \in T} cosine(s_i, w_j) \tag{4.4}$$

where $s_i$ denotes a synset vector while $w_j$ denotes a word vector. Note that same as WSD1, a threshold of 0.2 is set to filter unrelated words. WSD2 aims to evaluate the performance of joint vector representation of synsets and words.

**WSD3** implements the *Sim* function using synset-text similarity metric based on the composition property of word embeddings. In training, contextual words and synsets are used to predict the target word. Hence, the composition of contextual words may have a better vector representation than individual words. This is inherited from Word2Vec, which specifies special meaning to frequent collocations of words. In order to investigate such a property, we define $comb(T)$ function to derive the context vector by performing a normalized average on all the word vectors in text $T$, filtering those irrelevant words with a threshold of 0.2. Then, the disambiguation function is based on the cosine similarity of synset and text vectors:

$$Sim(s_i, T) = cosine(s_i, comb(T)) \tag{4.5}$$

WSD3 approach aims to evaluate the inter-connectivity between synset and composition of word vectors.

**WSD4** is an implementation of an unsupervised graph-based WSD (Sinha and Mihalcea, 2007; Navigli, 2009), which combines a synset-synset similarity measure and a graph centrality algorithm for WSD. The graph-based WSD method annotates all the words in a given sentence in a collective disambiguation way. Firstly, given a sequence of words with their corresponding candidate synsets, and for each word pairs in text, we compute their synset-synset similarity, obtaining an undirected synset similarity graph. Then, weighted PageRank (Sinha and Mihalcea, 2007) is performed in the similarity graph in order to rank all the synsets based on topical centrality. Finally, for each polysemic word in text, the candidate synset with the highest rank score is assigned as the correct sense of the target words. As shown later, WSD4 shows a performance of synset vector representation similar to that of WSD1, but using a different disambiguation approach.

| Dataset | Senseval-2 | Senseval-3 | SemEval-07 | SemEval-13 | SemEval-15 | SemCor | OMSTI |
|---|---|---|---|---|---|---|---|
| #Sents | 242 | 352 | 135 | 306 | 138 | 37176 | 813798 |
| #Annotations | 2282 | 1850 | 455 | 1644 | 1022 | 226036 | 911134 |

Table 4.2: Statistics of the Sense-Annotated datasets used in training and evaluation

### 4.1.4 Evaluation

We evaluate the Synset2Vec model in the task of fine-grained WSD (Alessandro Raganato and Navigli, 2017). Four unsupervised WSD systems based on synset and context similarity are used for evaluation in terms of measuring synset-synset similarity, synset-word similarity and synset-text similarity n the proposed joint semantic vector space of synsets and words. The goal of the experiments is to validate the following hypotheses:

- H1: The joint vector representation of synsets and words is effective.
- H2: Synset expansion and increasing the number of profiles can enhance the model.

We use the all-words fine-grained WSD evaluation datasets from (Alessandro Raganato and Navigli, 2017), as they uniformed the data format and sense inventory (WordNet 3.0) with several datasets, including Senseval-2 (Edmonds and Cotton, 2001), Senseval-3 (Snyder and Palmer, 2004), SemEval-07 (Pradhan et al., 2007), SemEval-13 (Navigli et al., 2013) and SemEval-15 (Moro and Navigli, 2015). Moreover, for enriching the synset profiles, we use the sense-annotated corpus SemCor (Miller et al., 1993), and a larger automatically constructed corpus OMSTI (One Million Sense-Tagged Instances) (Taghipour and Ng, 2015). Both datasets are publicly available. The statistics of evaluation dataset and training corpus are shown in Table 4.2, where #Sent and #Annotation denote the number of sentences and annotations, respectively. According to different amount of training data used for Synset2Vec model, we create four models *M1, M2, M3, and M4* for comparison: (1) M1: WordNet synset and gloss; (2) M2: M1 with synset expansion; (3) M3: M2 with SemCor; (4) M4: M3 with OMSTI. We then run these four models in four WSD systems respectively in order to validate our hypotheses. For the evaluation metric of WSD, we use the standard definitions of precision, recall and F-measure (Navigli, 2009). As the datasets are used for fine-grained WSD where each target word instance corresponds to a synset in WordNet, we only show the F-measure score for evaluation since the scores of precision, recall and F-measure are the same.

In addition, as we are experimenting with knowledge-based WSD, the implemented WSD systems with different models are compared against the Most Frequent Sense (MFS) computed from SemCor and WordNet first sense (WNFS). These two baseline systems are assumed to be difficult to challenge (Camacho-Collados et al., 2016). We also include a conventional semantic similarity metric (WN-JCN) using WordNet taxonomy and information content computed from SemCor (Jiang and Conrath, 1997) for comparison, which has been shown to yield the best performance in this task (Navigli, 2009). We run WN-JCN in WSD1 and WSD4. Moreover, we also use a word overlap based approach (Lesk, 1986) for comparison. Finally, as we are actually testing the word and synset inter-connectivity in a shared vector space, we also include results from a similar experiment from (Mancini et al., 2016), and denote them as SW2V (Mancini et al., 2016) and AutoExtend (Rothe and Schütze, 2015). Apart from the four Synset2Vec models, we also use a combination of WSD3, M4, and a back-off strategy (Camacho-Collados et al., 2016) (M4+WNFS) for comparing to SW2V, since such strategy is also included. We set the threshold value $\theta = 0.5$ as the confidence to decide if use the similarity measure or WordNet First Sense (WNFS).

Our implementation is based on NLTK[1] and Gensim[2], using Python. We use the CBOW model of Synset2Vec and set 200 dimensions for synset and word vectors with contextual window of 12. The evaluation results are shown in Table 4.3. In this table, we can see that the basic model M1 already obtains a fairly good performance when comparing to other systems, especially considering synset-synset relation (WSD1 and WSD4). M1 only contains information from WordNet itself, which shows the effectiveness of Synset2Vec. Note that WN-JCN only works for nouns and leaves other type of words with default WNFS. Furthermore, in M4+WNFS, the best F-scores are obtained comparing to other knowledge-based WSD systems, although it is still not as good as strong baselines MFS and WNFS. With such experimental observations, we validate our first hypothesis H1 that Synset2Vec is effective. Comparing M1, M2, and M3, the experimental results have shown that adding more profiles and performing synset expansion, the effectiveness of the models can be improved. This validates our hypothesis H2. However, comparison between M3 and M4 shows that automatically generated annotations (OMSTI) may not be as good as manually annotations (SemCor), due to the lower accuracy of the annotations. Comparing to SW2V and AutoExtend, although Synset2Vec does not rely on existing word embeddings from large corpora, it shows improvement in fine-grained sense representation based on WordNet and its annotation datasets. Moreover, the Synset2Vec does not train synset to synset relations like SW2V, but it obtains good performance in synset-synset similarity task based on vec-

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

[2] https://radimrehurek.com/gensim/

tors derived from related words. In other words, Synset2Vec actually obtains synset vector representation from words and related synsets.

Additionally, by investigating the difference between WSD3 and WSD4, we noticed that more annotation examples help to improve the synset-text performance. With larger training data, contextual words may appear frequently so that special word collocation patterns are more effective than individual words. This property of Synset2Vec is inherited from Doc2Vec, in which synsets and contextual words are used to predict target words together so that synset vectors memorize the meaning of frequent word patterns in context. This property is useful for applications requiring synset-text relation. In summary, through the experiments in WSD, we validate the hypotheses and show that the Synset2Vec model is effective in representing synsets and words in a shared semantic vector space, by illustrating the inter-connectivity of synset-synset, synset-word and synset-text.

| System | Senseval-2 | Senseval-3 | SemEval-07 | SemEval-13 | SemEval-15 |
|---|---|---|---|---|---|
| MFS | 65.6 | 66.0 | 54.5 | **63.8** | 67.1 |
| WNFS | **66.8** | **66.2** | **55.2** | 63.0 | **67.8** |
| Lesk | 50.6 | 44.5 | 32.0 | 53.6 | 51.0 |
| SW2V | - | - | 39.9 | 54,0 | - |
| AutoExtend | - | - | 17.6 | 31.0 | - |
| WSD1 + WN-JCN | **59.7** | **58.2** | **48.8** | **56.2** | **63.3** |
| WSD4 + WN-JCN | 55.3 | 51.9 | 45.1 | 55.1 | 61.9 |
| WSD1 + M1 | 43.4 | 37.2 | 29.2 | 43.3 | 44.4 |
| WSD1 + M2 | 43.2 | 39.6 | **34.7** | 47.1 | 45.5 |
| WSD1 + M3 | **46.0** | **42.8** | 30.8 | **47.4** | **48.5** |
| WSD1 + M4 | 41.6 | 37.9 | 28.1 | 43.1 | 46.6 |
| WSD2 + M1 | 44.6 | 37.8 | 30.5 | 45.3 | 46.3 |
| WSD2 + M2 | 46.6 | 42.9 | 32.3 | 48.7 | 50.1 |
| WSD2 + M3 | **50.2** | **45.9** | **33.6** | 48.8 | 52.3 |
| WSD2 + M4 | 48.7 | 44.0 | 33.4 | **49.8** | **53.7** |
| WSD3 + M1 | 45.2 | 38.1 | 30.8 | 46.2 | 46.1 |
| WSD3 + M2 | 45.4 | 43.1 | 32.5 | 49.1 | 49.5 |
| WSD3 + M3 | **50.1** | **45.1** | **35.4** | 49.6 | 52.2 |
| WSD3 + M4 | 50.0 | 44.7 | 34.9 | **50.9** | **54.1** |
| WSD4 + M1 | 43.7 | 38.1 | 29.5 | 43.8 | 44.8 |
| WSD4 + M2 | 44.3 | 40.1 | 31.9 | 46.9 | 46.9 |
| WSD4 + M3 | **48.9** | **42.5** | **34.5** | **48.0** | **50.0** |
| WSD4 + M4 | 43.8 | 39.3 | 33.2 | 45.1 | 49.4 |
| M4+WNFS | **61.2** | **58.9** | **49.2** | **58.2** | **63.6** |

Table 4.3: F-Measures percentage of different knowledge-based WSD systems and embedding models in five all-words fine-grained WSD datasets.

## 4.2 Named Entity Disambiguation

The increasing availability of LOD (Bizer et al., 2009a) has given birth to the notion of large scale KGs, with popular examples such as Freebase (Bollacker et al., 2008), DBpedia (Bizer et al., 2009b), and YAGO (Hoffart et al., 2013). Named Entity Linking (NEL) is a fundamental module for developing KG-based applications, including text analysis (Meij et al., 2012), document retrieval (Medelyan et al., 2008), knowledge base population (Ji and Grishman, 2011; Dredze et al., 2010), semantic search and question answering (Shekarpour et al., 2015). In general, a NEL system needs to detect a sequence of words (spots or mentions) in a given text, and to identify those mentions to entities registered in the given KG. The latter process of entity identification is not a trivial task because it needs to tackle two difficult problems, namely synonymy and polysemy. To address synonymy problem, a NEL system needs to match an entity despite its diverse name variations such as abbreviations, spelling variations, nicknames to name a few. The main approach to solve the synonymy problem is to construct entity name dictionaries as complete as possible in order to cover diverse name variations (Shen et al., 2015), and to apply approximate string matching (Dredze et al., 2010). Thus, the performance of these techniques is mainly concerned with the quality of name dictionaries and approximate matching algorithms. The polysemy problem is caused by the fact that multiple entities in KGs might have the same name, and this is quite common for named entities. The task of addressing the polysemy problem for named entities is called NED, and there is a large body of research techniques that have been proposed for addressing NED automatically (Cucerzan, 2007; Mihalcea and Csomai, 2007; Milne and Witten, 2008; Kulkarni et al., 2009; Mendes et al., 2011; Ganea et al., 2016). However, resolving the polysemy problem is a common challenge whose difficulty is equivalent to solving central problems of AI (Navigli, 2009). The accuracy of NED is far from perfect and related to many aspects of KGs, datasets and applications. This chapter focuses on researching of semantic similarity for unsupervised NED using most common semantic features that are available in most KGs, therefore, the proposed similarity-based disambiguation method can be conveniently applied to various KGs.

Current unsupervised NED approaches (Shen et al., 2015) are mainly based on local context (Hoffart et al., 2011), such as context similarity (Mendes et al., 2011), or global inference (Ratinov et al., 2011) using entity-entity relatedness (Milne and Witten, 2008). When the mention contexts are large text objects such as paragraphs or documents, where rich contextual information can be collected, the conventional context similarity approach is effective (Mihalcea and Csomai, 2007; Mendes et al., 2011; Hoffart et al., 2011). However, while processing short texts such as web queries, questions and tweets, limited contextual in-

| Entities | Word Features from Abstracts | Types |
|---|---|---|
| dbr:John_Noble | television, century, theatre, people, winner, actor, director, birth, performance, male, film, horror, fiction, role, series, action | dbo:Actor, yago:Director, dbc:Australian_Film_Actor |
| dbr:John_Noble(baritone) | baritone, singer, cancer, people, opera, title, favourite, role, composer, progress | yago:Artist, yago:Musician, dbc:English_opera_singers dbc:Operatic_baritones |
| dbr:John_Noble(bishop) | people, alumnus, bishop, lecturer, school, ministry, career, region, incumbency, teacher, position, chaplain | yago:Bishop, yago:Priest, dbc:Bishops_of_North_Queensland |
| dbr:John_Noble(painter) | painter, work, carving, landscape, gallon, canvas, photographer, exhibition, outbreak, picture, people, collection, child, post, artist | dbo:Artist, yago:Painter, yago:Creator dbc:20th-century_American_painter |

Table 4.4: Semantic Features of Candidate Entities for John Noble

formation may not be effective enough to discriminate ambiguous entities. According to the analysis of commercial search engines (Guo et al., 2009), less than 1% of the queries contain two or more named entities, while web queries and questions normally consist of few words (e.g. 3 words on average in search queries and 6-7 words on average in question queries). Obviously, when dealing with limited contextual information, entity-entity relatedness is no longer useful in handling single entity mention and context may not contain enough feature words for computing context similarity. For example, in a web question *"What movies did John Noble play"* (Berant et al., 2013), no other entities can help to discriminate the single ambiguous mention "John Noble". Table 4.4 shows some candidate entities of mention "John Noble" and their corresponding semantic features extracted from DBpedia (Bizer et al., 2009b) which is used as reference KG in this work because of its central role in LOD and various publicly available datasets. The words *movie* and *play* are contextual words, but they do not match the semantic features of the candidate entities. Although the word *movie* is obviously more similar to the entity *dbr:John_Noble* because of the feature words *actor, film, director*, the conventional context similarity can not address such fine-grained semantic closeness to identify the correct entity *dbr:John_Noble*. Consequently, we aim to exploit semantic similarity to develop disambiguation approach that can compare those words in different lexical forms but having similar meanings. In this way, semantic similarity is used to enhance the context similarity for NED when the contextual information is scarce and entity-entity relatedness is not available.

In order to make the similarity-based disambiguation approach applicable to various KGs, entity categories and textual descriptions are used as semantic features to apply semantic similarity methods for NED, because they contain rich semantic information and are available in most KGs. Based on the textual feature, we use IR (Baeza-Yates et al., 1999) and LSA (Deerwester et al., 1990) to develop the baseline of unsupervised NED approach based on context similarity through the computation of textual similarity between context and entity descriptions. Then we propose a novel Semantic Contextual Similarity based NED (SCSNED), which relies on contextual word similarity to improve the baseline that assumes equal importance of contextual words and provides coarse meaning comparison between context and entity descriptions. The SCSNED computes semantic similarity between individual words to offer fine-grained meaning comparison, and uses inverse entity frequency to consider the relative importance of feature words by counting word appearance in descriptions of candidate entities. In order to optimize the performance of SCSNED, we exploit the usage of both knowledge-based semantic similarity methods (Zhu and Iglesias, 2016) relying on semantic knowledge of WordNet (Miller, 1995), and corpus-based semantic similarity methods using word embedding model Word2Vec (Mikolov et al., 2013b) based on the statistical knowledge from textual corpus. Moreover, given that semantic categories are very effective in representing the meaning of entities (Bekkerman and Gavish, 2011), we propose a Category2Vec embedding model to compute word-category similarity for NED in order to provide complement to the word-word similarity feature. Category2Vec learns semantic category and word embedding jointly based on entity abstracts and entity categories, which treats those categories composed by multi-word expressions (e.g. Australian Film Actor) as a unique semantic unit without separating them into individual words. We found that word-category similarity based on the learned joint vector space is very effective for NED, while the learning of word and category vector representations are only depending on KGs themselves without labeled dataset.

In conclusion, this section proposes SCSNED method and exploits various similarity methods in the case of little contextual information and single entity mentions. The effectiveness of different similarity methods are identified through a comparative experiment on various datasets including web queries, web questions and tweets. Note that although we evaluate similarity methods with short texts, the proposed NED approach can be directly applied to larger text objects by decomposing them into sentences. Furthermore, we propose a Category2Vec model to compute word-category similarity that has been shown to be effective for NED. The experiments in tweet text have shown that combining baselines, SCSNED and Category2Vec methods have improved the state of art of unsupervised NED approaches. Moreover, unlike many current works optimize effective features for a particular KG, our focus is to exploit some common features and similarity methods that can be used in different kinds of KGs for the task of NED.

### 4.2.1 Background

In this section, we present the definition, scope, related works and the state of the art review for NED.

#### 4.2.1.1 The Definition, Scope and Related Works

Formally, given an input text consisting of a sequence of words $T = \{w_1, w_2, \ldots w_k\}$, a NEL system needs to recognize a set of entity mentions $M = \{m_1, m_2, \ldots m_n\}$ (called mention detection or entity recognition), and maps each entity mention $m \in M$ to a set of candidate entities $E_m$ which contains all possible entities registered in the given KG that have similar lexical surface form with the entity mention $m$ (called link generation or entity linking). When an entity mention $m$ has more than one entity candidate registered in the KG, $|E_m| > 1$, the NEL system needs to accurately select the correct entity $e \in E_m$ which is the most pertinent to describe the mention $m$. This process is referred as NED. Since NEL contains both entity recognition and disambiguation, sometimes it is also called Named Entity Recognition and Disambiguation (NERD) (Carmel et al., 2014a).

Named Entity Recognition (NER) is an important sub-task of Information Extraction Information Extraction (IE) in NLP research for many years. The task of NER is to detect entity mention from unstructured text and determine its categories such as person, location, organization to name a few. Thus, this important sub-task of IE is also called Named Entity Recognition and Classification (NERC) (Nadeau and Sekine, 2007). Many NER approaches have been proposed from early rule-based systems to recent systems employing machine learning techniques (Nadeau and Sekine, 2007). There are many publicly available NER tools such as Stanford NER (Finkel et al., 2005) which can be used directly. In many aspects, NER is closely related to NEL because both NER and NEL need to detect entity mention which is the reason of using NER as a precedence for NEL in some works (Hoffart et al., 2011). The main difference between NER and NEL is that NER classifies entity mention into predefined classes while NEL classifies entity mention into entities that are registered in a KB. The predefined classes usually have limited numbers while the number of entities in KG can usually reach to millions so that NEL has mention-entity mapping process in order to reduce the problem space to a limited number of entity candidates. In addition, the disambiguation task is slightly different. For example, when performing NER in queries (Guo et al., 2009), the disambiguation goal of NER is to classify entity mentions of *Harry Porter* into either class *book* or *movie* according to the given text, while NEL needs to annotate the entity mention with the correct registered entity in KG which can be either a book instance or movie instance. Although the disambiguation goal is different, since entities in KGs normally have a specific entity type (e.g. book or movie), recognizing the entity class can help to determine the corresponding entity in KGs. Therefore, recent researches (Guo et al., 2013; Sil and Yates, 2013) and challenges (Carmel et al., 2014b) are proposed to perform NER and NEL jointly. Designing models to represent the relations between entity context (surrounding words) and entity types is one of the main ideas to study NER and NEL jointly. For example, one recent related work (Guo et al., 2009) learns a probabilistic model of the semantic association between entity context and entity type from query log data using Latent Dirichlet Allocation (LDA), where entity context is treated as documents and entity class is treated as topics.

In this work, we do not consider the entity recognition problem and assume that entity mentions are given, which means we only consider the NED instead of the complete NEL system. For more technical details of NEL, of specific KG features, problem scopes, task assumptions, technical methods and performances, reader could refer to the complete NEL survey (Shen et al., 2015) and the NEL evaluation (Hachey et al., 2013; Cornolti et al., 2013). In addition, those entity mentions having no candidate entities recorded in the given KG, are defined as unlinkable mentions, such as $(m_i, NIL)$. Note that NIL is different from *pruning* (Piccinno and Ferragina, 2014a) which is used to discard detected mentions and their annotated entities if they are considered not interesting or pertinent to the semantic interpretation of the input text. We do not specially address NIL and *pruning* and assume that all the recognized entity mentions at least have one candidate entity recorded in the given KG. This assumption is similar to some of the works in *Wikification* such as Wikify (Mihalcea and Csomai, 2007).

The task of WSD (Navigli, 2009) is relevant to NED, because both WSD and NED need to address synonymy and polysemy problem. The task of WSD is defined as automatically assigning the correct sense of a polysemous word within a given context to a given sense inventory such as WordNet (Miller, 1995). For example, a WSD system needs to choose whether the polysemous word *bank* refers to a *repository for money* or a *pile of earth on the edge of a river* within a given context. Because of the similar task shared by NED and WSD, those previously proposed methods for WSD (Navigli, 2009) are applicable in NED. However, they also meet different challenges. In WSD, words need to be mapped to sense entry (in WordNet, denoted as synset consisting of a set of synonyms) based on lemma matching which has less lexical variations. In contrast, named entities are usually multi-words having more various surface forms which usually need to be mapped using fuzzy matching with larger name dictionaries. In addition, because of the stability of controlled vocabularies, WSD normally assumes that sense inventories such as WordNet are complete where a given word is assumed to be able to find its possible synsets. In contrary, KGs are continuously updated (e.g. new entries for new books or movies). Furthermore, NED addresses named entities which are modeled as instances in KGs, while WSD addresses common nouns (e.g. bank) which are usually treated as metaclasses in KGs indicating a group of instance. Therefore, WSD and NED are solving the disambiguation problem in different aspects. Nevertheless, they can be influenced by each other, because instance knowledge can help in class disambiguation while class knowledge is also useful in solving instance disambiguation. This has been shown in many recent researches such as Wikify (Mihalcea and Csomai, 2007) that uses Wikipedia as resource for WSD, and Babelfy (Moro et al., 2014) that solves WSD and NED jointly based on a wide-coverage semantic network.

Since we evaluate NED approaches with short texts such as web queries, web questions and tweets, we briefly review related works addressing short texts. Query segmentation separates queries into compound words or noun phrases that can be considered as individual concepts (Hagen et al., 2011; Pu and Yu, 2008) aiming to understand the correct query intent for document retrieval. NEL is a component of query understanding (Pound et al., 2012) over KGs for annotating entities in queries for further query classification (Shen et al., 2006) or query interpretation (Sawant and Chakrabarti, 2013). Hasibi et al. (Hasibi et al., 2016) exploit the NEL problem with entity retrieval problem jointly in order to improve the search performance. The efficiency problem of linking entities in queries has been studied in (Blanco et al., 2015) by introducing a probabilistic model, as well as hashing and compression techniques. This chapter only studies the effectiveness of NED in case of short texts without employing specific higher level applications. Moreover, NEL has been studied in many research works in case of microblog such as tweets (Meij et al., 2012; Guo et al., 2013; Liu et al., 2013; Shen et al., 2013) focusing on processing noisy and informal texts (Guo et al., 2013), user interest model (Shen et al., 2013) and entity filtering (Habib and van Keulen, 2015). We use tweet data as scenario of limited contextual information to evaluate the proposed NED approaches. The state of the art NED approaches are reviewed in the following section, while for techniques of NER in tweets, readers can refer to  (Ritter et al., 2011).

#### 4.2.1.2 The State of the Art

As ambiguous entity mentions have multiple candidate entities, various NED features and methods have been proposed and found to be effective during recent years. To present them clearly, we survey and compare the state of the art NED methods according to a general classification: (1) based on entity prominence; (2) based on context similarity; and (3) based on entity relatedness.

NED methods based on entity prominence select the most prominent entity for a given mention only based on the entity mention and the property of its candidate entities, without considering the surrounding context of the mention. Many prominence features have been used in NED systems including string similarity, popularity and commonness. String similarity is based on the name string comparison between mention and candidate entities using different similarity or distance metrics such as edit distance (Liu et al., 2013), Dice, Hamming distance (Dredze et al., 2010) to name a few. String similarity is the most straightforward and common feature for NED, but it is not reliable when candidate entities have same names or the mentions have many name variations. Popularity is another common prominence feature which is domain dependent and especially useful in the case of lacking contextual information, such as single entity mention in the query. Wikipedia page view statistics is a typical popularity feature that has been used to represent the entity popularity in many systems (Guo et al., 2013; Gattani et al., 2013). Another popularity feature is based on click popularity (Ji and Grishman, 2011). Both page view and click information are effective to select the most popular entity for individual ambiguous entity mentions when there is no other information to help discriminate candidate entities. However, both page view and click features are dependent on domain specific applications and will meet cold-start problem. Moreover, commonness (Medelyan et al., 2008) has been proven as a very effective prominence feature in many NED systems (Medelyan et al., 2008; Kulkarni et al., 2009; Hoffart et al., 2011; Ferragina and Scaiella, 2010; Ratinov et al., 2011; Shen et al., 2012a; Guo et al., 2013; Liu et al., 2013). The entity commonness denotes the prior probability of entity which is computed from sense distribution over entity annotation corpora such as anchor text of Wikipedia. If a word or n-gram $a$ appears as an annotation in corpora $N$ times and there are $m$ times linking to the entity $E$, then the *commonness* of entity $E$ can be computed as $P(E|a) = \frac{m}{N}$. Computing entity commonness is dependent on entity annotation corpora which is difficult to obtain and the computed entity commonness probability may only have limited entity coverage because of the incompleteness of the annotation corpora.

NED methods based on context similarity discriminate ambiguous entities through measuring similarity between the mention context and the candidate entities. Context similarity metrics depend on different semantic features in representing contexts and entities. The most intuitive semantic features to represent context are different granularity of texts surrounding the mention, from whole input text to several surrounding words. Similarly, entities can be represented by the textual descriptions extracted from KGs, ranging from the entire Wikipedia page (Bunescu and Pasca, 2006), paragraphs of Wikipedia page (Kulkarni et al., 2009; Mendes et al., 2011), entity summaries (Ratinov et al., 2011), entity abstracts (Meij et al., 2011), entity categories (Bunescu and Pasca, 2006), entity types (Guo et al., 2013), keyphrases (Hoffart et al., 2011), entity titles (Liu et al., 2013), and anchor texts (Kulkarni et al., 2009) to name a few. Then, context-entity similarity can be computed with different similarity metrics based on different models for textual features. The simplest model is BOW where both contexts and entities are represented as set of words, concepts or keyphrases (Hoffart et al., 2011). In consequence, the context-entity similarity is computed based on set intersection that counts the overlap of words, concepts, keyphrases (Hoffart et al., 2011) between context and entity (Mihalcea and Csomai, 2007) similar to the idea of Lesk algorithm (Lesk, 1986) for WSD. Common metrics to compute such overlap similarity are Jaccard similarity or Dice coefficient (Mihalcea and Csomai, 2007; Bunescu and Pasca, 2006; Liu et al., 2013; Kulkarni et al., 2009; Hoffart et al., 2011). Apart from the simple BOW, VSM has been used to represent contexts and entities into high dimensional context and entity vectors, whose values of each dimension are Term Frequency (TF)-IDF scores (Baeza-Yates et al., 1999) computed from specific text collection and particular vocabulary. Then the context-entity similarity is computed using dot-product or cosine similarity between context and entity vectors (Bunescu and Pasca, 2006; Kulkarni et al., 2009; Dredze et al., 2010; Han et al., 2011; Mendes et al., 2011; Ratinov et al., 2011; Guo et al., 2013; Milne and Witten, 2013). Moreover, some recent works proposed to learn distributed vector representation of mention, context and entity for NED (He et al., 2013; Sun et al., 2015; Francis-Landau et al., 2016) with deep learning architecture (Hinton et al., 2006), and proposed to extend the contextual words with similar words (Blanco et al., 2015) based on Word2Vec (Mikolov et al., 2013b). In addition, probabilistic language models (Han and Sun, 2011; Meij et al., 2011) and topic models (Pilz and Paaß, 2011; Kataria et al., 2011; Houlsby and Ciaramita, 2014) have been applied to model context, mention and candidate entity in in order to rank the entities given the specific context and mention. The context and entity co-occurrence knowledge are encoded to compute context-entity similarity according to the probabilistic likelihood of an entity appearing in a specific context. Following such idea, recent NED works formulate probabilistic framework considering various type of statistical information including mention-entity probability, entity-entity co-occurrence, contextual word-entity statistics from entity annotation dataset (e.g. Wikipedia anchor dataset) (Blanco et al., 2015; Ganea et al., 2016).

Entity relatedness is a special case of context similarity, since entities of other mentions in the input text are used as the semantic feature to represent context. According to the assumption that the input text contains coherent entities from one or few related topics (Hoffart et al., 2011), multiple ambiguous entities are discriminated collectively (Kulkarni et al., 2009) based on entity relatedness. Such collective disambiguation model is a global model that discriminates all entity mentions jointly (Ratinov et al., 2011). In contrast, NED methods based on entity prominence and context similarity use frequently a local model (Ratinov et al., 2011) which considers each entity mention in isolation. Collective disambiguation of all entity mentions in an input text is shown as NP-hard (Hoffart et al., 2011) problem, which is usually simplified by comparing unambiguous entities with ambiguous entities (Cucerzan, 2007), combining with entity prominence (Ferragina and Scaiella, 2010), or averaging the coherence (Shen et al., 2012a). In fact, most of NED methods combine both local and global models to achieve better disambiguation performance (Ratinov et al., 2011). The key module of this collective disambiguation model is measuring entity relatedness in order to infer the coherence among candidate entities for all mentions. There is a number of semantic features that can be used to compute entity-entity relatedness based on different type of information sources. Firstly, semantic contents of entities such as textual descriptions and semantic categories are represented in BOW or VSM to compute entity-entity similarity based on: (1) dot or cosine similarity of entity description or category vectors (Cucerzan, 2007); (2) topical coherence between entities using overlap of weighted keyphrases (Hoffart et al., 2012b) and topic models (Piccinno and Ferragina, 2014a); and (3) semantic similarity of entity category hierarchies (Shen et al., 2012a). Secondly, from entity annotated corpora, entity co-occurrence (Nunes et al., 2013) and entity distribution (Aggarwal et al., 2015; Shen et al., 2012b) are used to compute entity-entity relatedness based on the application of distributional hypothesis (Turney et al., 2010) which assumes that entities occur in similar contexts are semantically related. Finally, apart from semantic content analysis and distributional analysis, graph analysis is also very effective in measuring entity connectivity in order to compute entity-entity relatedness, given that entities are connected to each other in KGs. Graph analysis measures the entity relatedness based on semantic entity networks using degree analysis (Milne and Witten, 2008) or relational analysis (Hulpuş et al., 2015). Degree analysis counts the edges connecting entities which only represent occurrence, incoming, or outgoing information, while relational analysis considers semantic meaningful relations between entities. This difference results in different kind of entity relatedness methods. Milne and Witten (Milne and Witten, 2008) proposed a degree analysis method for computing entity relatedness based on the incoming and outgoing links, which is similar to the Normalized Google Distance (Cilibrasi and Vitanyi, 2007). This entity relatedness method has been popularly adopted by many subsequent NED systems (Medelyan et al., 2008; Kulkarni et al., 2009; Han and Zhao, 2009; Ferragina and Scaiella, 2012; Hoffart et al., 2011; Han et al., 2011; Ratinov et al., 2011; Shen et al., 2012a; Liu et al., 2013). Following a similar idea, some variations based on degree analysis include Pointwise Mutual Information (Ratinov et al., 2011) and Jaccard distance (Guo et al., 2013). Recent works started to consider semantic relations between entities in KG and use relation analysis for computing entity relatedness based on the shortest path between entities (Nunes et al., 2013) and relation weighting in the shortest path (Hulpuş et al., 2015).

With various entity relatedness methods, it is obvious that the performance of entity relatedness can be further optimized and enhanced by combining different methods through supervised machine learning techniques (Ceccarelli et al., 2013). Similarly, better NED performance can be achieved by combining different disambiguation methods and features, and using a labeled dataset to learn to assign proper entities with supervised learning methods, such as naive bayes classier (Mihalcea and Csomai, 2007), support vector machine (Bunescu and Pasca, 2006; Ratinov et al., 2011; Meij et al., 2011), and learning to rank framework (Milne and Witten, 2008) to name a few. As preparing labeled datasets requires tremendous efforts, when labeled datasets are not available, unsupervised NED approaches are needed. A simple but effective unsupervised disambiguation method is to select the correct entity with the highest similarity score computed based on entity prominence, context similarity and entity relatedness (Medelyan et al., 2008; Ferragina and Scaiella, 2010; Mendes et al., 2011; Han and Sun, 2011; Shen et al., 2012a). Apart from similarity-based methods, more complicated unsupervised disambiguation method is graph-based approach that combines various disambiguation features into graph representation. Specifically, mention, context and entity are modeled as nodes in graph, while their semantic associations are modeled as edges. Then, various graph-based algorithms (Piccinno and Ferragina, 2014a) can be applied to make disambiguation decisions, such as PageRank (Hachey et al., 2011), Personalised PageRank (Han et al., 2011), dense subgraph estimation (Hoffart et al., 2011), degree-based importance measure (Guo et al., 2011), Hypertext-Induced Topic Search (HITS) (Usbeck et al., 2014) and many others. Various disambiguation features and methods described above represent different aspects and consideration in dealing NED. No feature or method is superior than others over all kinds of datasets (Shen et al., 2015). Thus, disambiguation features and methods need to be selected according to the specific characteristic of dataset and the requirement of application in tradeoff between precision and recall, accuracy and efficiency. With little contextual information, we propose to use word-word and word-category similarity to enhance the computation of context-entity similarity.

#### 4.2.1.3 Overview of Semantic Similarity

Semantic similarity methods give numerical similarity scores to words in order to represent their semantic distance. We have presented the state of the art semantic similarity methods in previous chapters. In this section, we briefly compare corpus-based methods and knowledge-based methods (Zhu and Iglesias, 2016).

Corpus-based semantic similarity methods are based on word associations learned from large text collections following the distributional hypothesis (Turney et al., 2010). Two words are assumed to be more similar if their surrounding contexts are more similar or they appear together more frequently. The computation of corpus-based methods are based on statistics of word distributions or word co-occurrences. We use word embedding tools Word2Vec (Mikolov et al., 2013b) to learn dense vector representation of words, because Word2Vec has been reported to have good performance in many applications (Baroni et al., 2014) and our proposed Category2Vec model is based on it. As suggested by the Word2Vec authors (Mikolov et al., 2013b), the CBOW model is more computationally efficient and suitable for larger corpus than the skip-gram model. Thus, the CBOW model is used to train word vectors in a neural network architecture which consists of an input layer, a projection layer, and an output layer to predict a word given its surrounding words with a certain context window size. Having the trained word vectors (the dimension is predefined empirically and we set 300 in our experiments), word similarity are computed using standard cosine similarity. Furthermore, due to the simple neural network architecture and the use of hierarchical softmax, Word2Vec is able to address large corpus and the training is very efficient. However, since the training of word vectors only use word sequences, a wide variety of word relations are considered as equally related according their co-occurrences, which makes the similarity between trained word vectors coarse and unable to address synonymous words and hierarchical relations accurately. In consequence, knowledge-based semantic similarity methods are considered to enrich some commonsense knowledge of words.

Knowledge-based semantic similarity methods measure the semantic similarity between words based on an ontology. Two words are considered to be more similar if they are located closer in the given ontology. The lexical database WordNet (Miller, 1995) is used as background ontology. Knowledge-based semantic similarity methods are designed to encode structural information of ontology to improve semantic similarity between words. Many knowledge-based methods have been proposed in the literature (Budanitsky and Hirst, 2006) for measuring similarity in WordNet exploiting various information such as shortest path length, depth, and IC. We select the Path (Rada et al., 1989) method, Wu & Palmer (Wu and Palmer, 1994) method, Resnik (Resnik, 1995) method, Lin (Lin, 1998) method, Jiang

| | Knowledge-Based WPath Similarity | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| words | movie | play | singer | teacher | painter | composer | opera | theatre |
| film | 1.000 | 0.660 | 0.160 | 0.143 | 0.295 | 0.143 | 0.218 | 0.251 |
| actor | 0.150 | 0.125 | 0.544 | 0.252 | 0.296 | 0.252 | 0.135 | 0.160 |
| baritone | 0.157 | 0.230 | 0.839 | 0.158 | 0.174 | 0.158 | 0.204 | 0.157 |
| director | 0.113 | 0.010 | 0.416 | 0.174 | 0.587 | 0.728 | 0.105 | 0.118 |
| bishop | 0.157 | 0.251 | 0.174 | 0.158 | 0.174 | 0.158 | 0.143 | 0.157 |
| picture | 1.000 | 0.660 | 0.68 | 0.123 | 0.251 | 0.113 | 0.230 | 0.218 |
| photographer | 0.123 | 0.100 | 0.471 | 0.194 | 0.681 | 0.587 | 0.113 | 0.130 |

Table 4.5: Word Similarity scores computed by WPath

& Conrad (Jiang and Conrath, 1997) method, and WPath (Zhu and Iglesias, 2016) method to compute semantic similarity between words based on WordNet. The details of those knowledge-based methods have been presented in previous chapters, while we show illustrative comparison between knowledge-based methods and corpus-based methods in this section.

Corpus-based and knowledge-based methods have different pros and cons in measuring word similarity. Corpus-based methods usually have better coverage of vocabulary because their computational models can be effectively applied to various and updated corpora. In case of KGs, since many entity descriptions normally contain domain specific terms which are not covered in common sense dictionaries such as WordNet, corpus-based tools like Word2Vec can capture domain specific vocabulary. On the other hand, because corpus-based methods do not consider different word meanings and various word relations, the learned word vectors are not as accurate as knowledge-based methods in some cases when words have special relations. For example, as illustrated in Table 4.5 and Table 4.6, *movie* and *film* are synonyms so they have highest similarity score of one in knowledge-based methods, while *baritone* is a sub-concept of *singer* so they should be more similar than *actor* and *singer*. Furthermore, since the main semantic information used by corpus-based methods are word sequence statistics from corpora, when the training corpora change, the word vectors would change and the similarity between words are different. While knowledge-based methods rely on ontologies which are normally fixed and stable, word similarity scores are different only when the corresponding similarity metric changes. In Table 4.5, we have

| | Corpus-Based Word2Vec Similarity | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| words | movie | play | singer | teacher | painter | composer | opera | theatre |
| film | 0.751 | 0.170 | 0.224 | 0.079 | 0.105 | 0.236 | 0.179 | 0.310 |
| actor | 0.401 | 0.168 | 0.550 | 0.246 | 0.356 | 0.423 | 0.236 | 0.314 |
| baritone | 0.054 | 0.147 | 0.450 | 0.224 | 0.259 | 0.497 | 0.42 | 0.123 |
| director | 0.200 | 0.037 | 0.225 | 0.308 | 0.150 | 0.345 | 0.131 | 0.265 |
| bishop | 0.000 | 0.032 | 0.083 | 0.205 | 0.124 | 0.154 | 0.062 | 0.058 |
| picture | 0.407 | 0.085 | 0.072 | 0.050 | 0.142 | 0.066 | 0.029 | 0.142 |
| photographer | 0.180 | 0.000 | 0.387 | 0.381 | 0.567 | 0.440 | 0.082 | 0.123 |

Table 4.6: Word Similarity scores computed by Word2Vec

shown the WPath (Zhu and Iglesias, 2016) method computing word similarity based on WordNet, while Word2Vec model trained from Wikipedia dump is shown in Table 4.6. We will compare different knowledge-based methods and Word2Vec model with different corpora in our experiment.

In addition, comparing rows and columns of Table 4.5 and Table 4.6, both types of similarity methods have given the same rank orders to those word pairs through their similarity scores, whereas some cases also show the difference between two kinds of methods. Two main reasons might have caused such a difference. Firstly, knowledge-based methods mainly study the concept taxonomy of WordNet, thus they are preferred to give higher similarity to those concepts in the same branch of the taxonomy, and give lower similarity to related words, such as *actor* and *movie*. Secondly, many common sense knowledge is usually not described so it is not contained in many textual corpus. In this case, corpus-based method may not be able to represent it. For example, there are some zero similarity values in corpus-based methods and the word *play* is given low similarity to *film* and *actor*. In summary, considering those pros and cons of both types of methods, it is better to combine both for NED in a given domain.

### 4.2.2 Semantic Contextual Similarity for NED

In this section, we present baseline approaches, and propose SCSNED and Category2Vec approach for unsupervised NED based on semantic contextual similarity.

#### 4.2.2.1 The Baseline Approaches

As entity descriptions are common and effective textual features available for most of KGs, IR techniques (Baeza-Yates et al., 1999) have been applied in many NED systems (Bunescu and Pasca, 2006; Mihalcea and Csomai, 2007; Mendes et al., 2011) to compute text similarity scores to discriminate the ambiguous candidates. We use them as the baseline approach of NED based on context similarity.

Context similarity is based on measuring vector similarity over standard VSM for mention context and entity descriptions, where both contexts and entities are represented as high dimensional vectors $v \in \mathbb{R}^{|V|}$. Each dimension of the vector $v$ corresponds to a word in the vocabulary $V$ which is created from all the entity descriptions in KGs. When the vocabulary is created, lemmatization is applied and those stop words, too frequent words and too rare words are filtered based on application requirements. In our illustrative example of DBpedia (see Table 4.4), words that appear less than 20 times and occur in descriptions of more than 50% entities have been removed, which has resulted in a vocabulary $|V| = 100000$. The value in each dimension of vector $v$ is represented by the corresponding word weight and computed using standard TF and IDF (Baeza-Yates et al., 1999). Formally, $tf(w_i, d)$ denotes the frequency of word $w_i$ in the document $d$, while $df(w_i)$ denotes the document frequency of the word $w_i$ which is numbers of entities whose textual descriptions contain the word $w_i$. The weight of dimension $i$ of document $d$ is defined as the product of TF and IDF of word $w_i$:

$$v_{i,d} = tf(w_i, d) * (1 + log\frac{N}{1 + df(w_i)}), \tag{4.6}$$

where $N$ is the total number of entities in KG and word $w_i$ corresponds to a token in the vocabulary $V$. If the word $w_i$ in vocabulary $V$ is not contained in a particular document $d$, then $tf(w_i, d) = 0$ and $v_{i,d} = 0$. Given a context vector $v_c$ and an entity description vector $v_{e_i}$, the cosine similarity is the cosine of the angle between vectors of context and entity:

$$sim_{cos}(v_c, v_{e_i}) = \frac{v_c \cdot v_{e_i}}{\|v_c\|_2 \times \|v_{e_i}\|_2} \tag{4.7}$$

The cosine similarity between context vector $v_c$ and entity vector $v_{e_i}$ can be viewed as the degree of correlation between words from mention context and entity description. Since mention context and candidate entities usually only contain a few words from vocabulary, the vector $v$ is normally a sparse vector so the correlation may be very low when entity descriptions do not contain many words appearing in the mention context. The vocabulary mismatch problem would result in failure of measuring similarity between context and entities when they do not contain same words in the vocabulary. Furthermore, the construction of vector $v$ only counts those words appearing in contexts or entities without considering

| Model | dbr:John_Noble | dbr:John_Noble(baritone) | dbr:John_Noble(bishop) | dbr:John_Noble(painter) |
|---|---|---|---|---|
| TFIDF | 0.0 | 0.0 | 0.0 | 0.0 |
| LSA | 0.355 | 0.094 | 0.0075 | 0.0874 |

Table 4.7: Context-Entity Similarity Based on TFIDF or LSA

their related words semantically.

In order to overcome the sparseness and vocabulary mismatch problem in standard IR-based text similarity model, a topic model LSA (Deerwester et al., 1990) is used to group similar words into latent topics through dimension reduction. Higher dimensional TF-IDF vectors are transformed into lower dimensional dense vectors in which each dimension denotes a latent topic. With those latent topics, even if the contextual words are not occurred in the description of candidate entities, the occurrence of their synonyms or related words can be counted as meaningful evidence to indicate the semantic relevance between contexts and entities. This is achieved by measuring similarity between context and entities in a second order relations among words. LSA operates SVD on the TF-IDF word-entity matrix $M$ of vocabulary $V$ and $N$ entity descriptions from KG. The semantic representation is obtained from word co-occurrence information by discovering latent topics and using them to represent contexts and entities. Formally, SVD factors the matrix $M$ into three matrices according to the following equation:

$$M_{|V|\times N} = U_{|V|\times K}\Sigma_{K\times K}S^{T}_{K\times N} \tag{4.8}$$

where $\Sigma_{K\times K}$ is the diagonal $K \times K$ matrix containing the $K$ singular values and $U$ and $S$ are orthogonal matrices. $|V|$ and $N$ are the number of words and entities. Typically we can remove some insignificant dimensions by retaining only the $K^{'}$ largest singular values in $\Sigma$ and setting the remaining small ones to zero. The original $M$ is approximated by $K^{'}$ largest singular triplets and the new vector space becomes the latent semantic topical vector space. In consequence, the original context and entity vector $v$ in standard VSM can be transformed into $K^{'}$ dimensional topic vectors through the following equation:

$$\hat{v} = v^{T}U_{|V|\times K'}\Sigma^{-1}_{K'\times K'} \tag{4.9}$$

With the lower dimensional topic vectors of context and candidate entities ($K^{'} = 300$ is chosen in this work), the context similarity is also implemented through cosine similarity defined in Eq.(4.7).

In summary, LSA creates a vector space model with latent topics rather than vocabulary $V$, and enables a homogeneous representation of words, sentences and documents. Because of this, LSA is able to address vocabulary mismatch problem between contextual words and entity descriptions, and give meaningful similarity scores to rank the candidate entities. For an illustrative example of entity mention *John Noble*, Table 4.7 shows the text similarity scores between contextual word *movie* and its candidate entities based on standard IR and LSA respectively. The example shows that the TF-IDF has failed in ranking candidate entities since it gives zero score to every candidate when *movie* has not occurred in all the entity descriptions. In terms of LSA, since word *movie* and entity descriptions are mapped into latent topical vector space, all the candidates have been assigned a similarity score, while the expected entity *dbr:John_Noble* has been given higher text similarity score than other entities. The example demonstrates that LSA handles better the vocabulary mismatch over TF-IDF. In addition, both IR and LSA model the context-entity similarity using text similarity, which has coarse-grained semantic meaning representation since text vectors are composed from multiple words. In the following sections, we present a fine-grained meaning representation using semantic similarity between words.

#### 4.2.2.2 The Word Similarity Approach

When contextual information is limited, contextual words are key evidences for NED, therefore more fine-grained semantic similarity models are critical to quantify the relevance between context and candidate entities through word similarity.

Semantic association between contextual words and candidate entities can help to select the proper entity. Since both textual descriptions and categorical labels of entities contain informative words to represent candidate entities, the word-entity similarity can be measured through word-word similarity between contextual words and entity feature words. For example, the entity *dbr:John_Noble* has the textual description “John Noble (born 20 August 1948) is an Australian film and television actor...", and categorical labels from Wikipedia categories *1948 births Australian male film actors*. Meaningful feature words can be extracted from those textual descriptions. For illustration, the example of extracted feature words of candidate entities for the mention *John Noble* are shown in Table 4.4. Note that nouns are usually more informative than verbs, adjectives and adverbs for NED so we mainly consider nouns as feature words of an entity. Words such as *actor, film, director* in the entity *dbr:John_Noble* are more relevant to the word *movie* than other words in other entities such as *baritone, singer, bishop, school, painter, photographer.*

Algorithm 2 outlines the details of SCSNED approach for disambiguation. Given the mention context and a set of entity candidates, the approach tries to identify the correct entity for the given mention. The functions $words(context)$ and $words(entity)$ retrieve the

**Algorithm 2** The SCSNED Approach for Disambiguation

```
procedure DISAMBIGUATE(context, candidates, K)
    $C \leftarrow words(context)$
    $E \leftarrow candidates$
    $ef \leftarrow frequency(word \in candidates)$
    $score \leftarrow 0, entity \leftarrow \emptyset$
    for all $e \in E$ do
        $F(e) \leftarrow words(e)$
        for all $w \in C$ do
            for all $f \in F(e)$ do
                $S \leftarrow sim_{word}(w, f) * (1 + log\frac{|E|}{1+ef(f)})$
            end for
        end for
        $value \leftarrow sum(top(S, K))$
        if $value > score$ then
            $entity \leftarrow e$
        end if
    end for
    return $entity$
end procedure
```

feature words for context and entity respectively. In order to quantify this similarity model between words, a word similarity function $sim_{word}(w_i, w_j) \in [0, 1]$ is used to give numerical score of the similarity between word $w_i$ and $w_j$. Formally, BOW is used to represent both contexts and candidate entities while semantic similarity is used to compare individual items in two sets semantically, instead of lexical matching. Given a set of entity candidates $E = \{e_1, \ldots, e_n\}$ for mention $m$, $F(e_i)$ receives the feature words of a candidate entity $e_i$ and $C$ receives a set of feature words in the surrounding context of mention $m$. Then the word similarity scores between words in context and words in candidate are computed. The top $K$ similarity scores are selected and summed to generate a weight value as being the correct entity, which is shown as $value \leftarrow sum(top(S, K))$ in Algorithm 2. This similarity computation is repeated over all the candidate entities, while the entity with highest weight value is returned as correct entity. The formal definition of SCSNED approach is shown in the following function:

$$\hat{e} = \underset{e_i \in E}{argmax} \sum_{w_i \in C, w_j \in F(e_i)}^{K} sim_{word}(w_i, w_j) * (1 + log\frac{|E|}{1 + ef(w_j)}) \tag{4.10}$$

where $|E|$ is numbers of candidate entities and $ef(w_j)$ counts numbers of candidate entities containing word $w_j$. $\sum^K$ is a sum function that selects and sums the top-K word similarity scores. The idea of $ef(w_j)$ is similar to document frequency for computing IDF. Given that some feature words may occur frequently in entity descriptions (e.g. word *person* occurs in every candidate entity of *john noble*), $ef(w_j)$ is used to give lower weight to those less discriminative words. $K$ is designed as parameter that can be determined empirically or optimized from datasets. If $K$ is set to smaller values such as one, two or three, the SCSNED may not able to discriminate multiple candidate entities because they may contain several feature words having same similarity scores. On the other hand, if $K$ has been set a bigger value, too much irrelevant feature words having lower similarity scores may be included in ranking so that the ranking precision would be affected.

Furthermore, the SCSNED becomes a semantic ranking model relying on semantic similarity of meaningful words, which is more accurate than text similarity in addressing synonymous and polysemous words. Since users in different backgrounds will describe the same information using different words (e.g. *movie* and *film* are synonymous words), semantic similarity can solve this problem by giving higher similarity scores to those semantically equivalent but lexically different words. Moreover, semantic similarity can partially solve polysemous word problem since it always gives highest similarity score between two words representing their closest meaning. Because of the important role of semantic similarity methods for words, we exploit the usage of both corpus-based and knowledge-based semantic similarity methods for the SCSNED approach in evaluation.

#### 4.2.2.3 The Category2Vec Approach

According to near-sufficiency property (Bekkerman and Gavish, 2011), semantic categories are informative to represent the meaning of an entity (e.g. director, actor) which usually contain more information about entity than longer entity descriptions. For example, as shown in Table 4.4, a candidate entity *dbr:John_Noble* has semantic categories such as yago:Director, dbc:Australian_Film_Actor. In the previous section, we have discussed the word similarity method based on meaningful words extracted from decomposing categories into individual words. The decomposition process may lose specific meaning of multi-word expressions. For example, the category dbc:Australian_Film_Actor has a more specific meaning indicating a specific group of actors, than those individual words of Australian, film and actor. Moreover, many current works (Bunescu and Pasca, 2006; Cucerzan, 2007; Shen et al., 2012a) have studied semantic categories such as Wikipedia categories for entity disambiguation. However, to the best of our knowledge, current works mainly focus on category to cate-

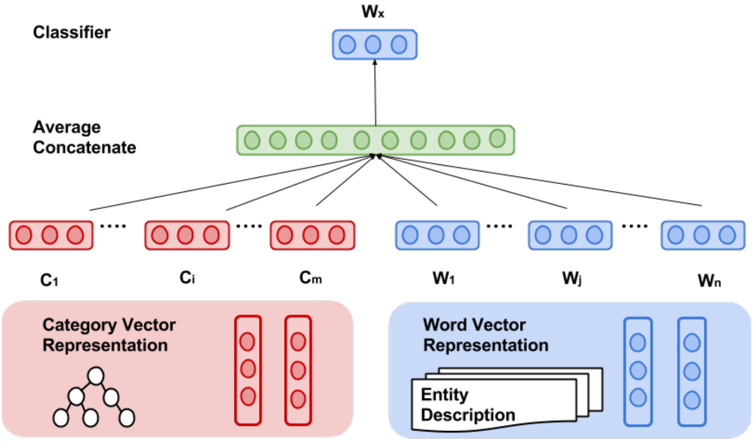


Figure 4.2: Jointly Learning Embeddings of Word and Category through Entities in KG

gory similarity (Bunescu and Pasca, 2006; Shen et al., 2012a) for disambiguation (Kulkarni et al., 2009), which would fail in single entity mention. Considering those problems in using semantic categories, we aim to develop word-category similarity to be the complement of word-word model in order to retain the complete meaning of categories.

Semantic categories can be viewed as semantic tags annotating entity descriptions, providing meaningful abstract keywords to entity descriptions. The co-occurrence of categories and words in entity descriptions can be used to learn the word-category associations. Applying the distributional semantics hypothesis (Turney et al., 2010), a word and a category are assumed to be more similar if they appear together more frequently. Following this idea, Word2Vec (Mikolov et al., 2013b) can be used to learn words and categories embedding by treating categories as special tokens appearing together with words. Then the similarity between word and category can be computed by cosine similarity of their vectors in shared vector space. We use Doc2Vec (Le and Mikolov, 2014) to build Category2Vec model for training category and word embedding jointly. Doc2Vec is a generalization of Word2Vec model to go beyond word-level to achieve phrase level or sentence level of distributed vector representation. In Doc2Vec, variable length sized text such as documents, paragraphs and sentences are treated as special words and trained with normal words jointly on top of Word2Vec learning framework. Those documents, paragraphs, and sentences act as memory for recording the main topics of co-training words. Correspondingly, in Category2Vec, categories are treated as special words recording the main topic of entity description. The Category2Vec learning framework is illustrated in Figure 4.2. Formally, given a sequence of training words $\{w_1, w_2, \ldots w_T\}$ from an entity description, and a sequence of semantic categories $\{c_1, c_2, \ldots c_j\}$ denoting the categorical feature of entity, word vectors and category vectors are trained jointly into a same distributed vector space by maximizing the average log probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq k\leq c,k\neq 0} logp(w_t|c_1, c_2, \dots c_j, w_{t-k}, \dots, w_{t+k}) \tag{4.11}$$

where $p(w_t|c_1, c_2, \dots c_j, w_{t-k}, \dots, w_{t+k})$ is the hierarchical softmax of the word vectors and category vectors (Le and Mikolov, 2014), while $k$ is the context window size. The category vectors and word vectors are trained using stochastic gradient descent and the gradient is obtained via back propagation (Rumelhart et al., 1988). The category vectors contribute to the prediction task of the next word given contextual words sampled from entity description. The contextual words are fixed-length and sampled from a sliding window over all the entity descriptions in a given KG. As shown in Figure 4.2, in Category2Vec learning framework, every category and word are mapped to unique vectors and their average or concatenation are used to predict the next word in a context. The word vectors are shared across all the entities as long as the word occurs in the entity description, while the category vectors are shared across those entities having the same category. As training results, we can obtain both word and category vectors, while word vectors are equivalent to Word2Vec model described in previous section.

In trained Category2Vec model, since word and category vectors are in the same shared vector space, they can be directly used to compute word-category similarity based on cosine similarity of vectors. We use two strategies to develop similarity based NED, namely *max* and *average* strategies. With the *max* strategy, we treat categories as special feature words and use the following function to implement NED approach.

$$\hat{e} = \underset{e_i\in E}{argmax} \sum_{w_i\in C, f_j\in F(e_i)}^{K} sim_{category2vec}(w_i, f_j) \tag{4.12}$$

where $f_j \in F(e_i)$ denotes the categories of entity $e_i$, and the similarity function $sim_{category2vec}$ is cosine similarity between word and category in the joint vector space learned from Category2Vec. In this strategy, Category2Vec is used as another corpus-based similarity method while it computes word-category similarity. In the *average* strategy, contextual words and entity categories are first mapped to corresponding vectors, and then combined respectively with normalized average. Then, the context and entity similarity is computed based on cosine similarity between averaged context and entity vector.

$$\hat{e} = \underset{e_i\in E}{argmaxsim_{cosine}}(avg(C), avg(F(e_i))) \tag{4.13}$$

where $avg(C)$ and $avg(F(e_i))$ are the normalized average vectors of contextual words and categories. This strategy is actually simulating the computation of text similarity. We will compare *max* strategy and *average* strategy in our experiment.

| dbc:Artificial_intelligence | | | |
|---|---|---|---|
| Category | Similarity | Word | Similarity |
| dbc:Machine_learning | 0.761 | application | 0.613 |
| dbc:Cognitive_science | 0.709 | process | 0.581 |
| dbc:Information_retrieval | 0.658 | intelligent | 0.575 |
| dbc:Semantic_Web | 0.658 | analysis | 0.565 |
| dbc:Artificial_neural_networks | 0.654 | methodology | 0.563 |
| dbc:Data_modeling | 0.652 | algorithm | 0.552 |
| dbc:Knowledge_engineering | 0.651 | logic | 0.547 |
| dbc:Automated_planning_and_scheduling | 0.638 | knowledge | 0.543 |
| dbc:Information_systems | 0.635 | simulation | 0.543 |
| dbc:Learning | 0.631 | interaction | 0.528 |
| dbc:Semantics | 0.629 | learn | 0.521 |
| dbc:Natural_language_processing | 0.628 | cognition | 0.505 |
| dbc:Decision_theory | 0.624 | communicate | 0.501 |
| dbc:Simulation | 0.609 | heuristic | 0.500 |
| dbc:Model_checkers | 0.608 | mathematic | 0.482 |

Table 4.8: Examples of top-15 similar categories and words of dbc:Artificial_intelligence

Moreover, in the Category2Vec model, since categories are treated as special words for recording the semantic topics of a group of words (concatenation of textual words from list of entity descriptions), compared to latent topics in topic models (Kataria et al., 2011), categories can be viewed as explicit topics representing human defined domain knowledge. Comparing to entity embedding (Zwicklbauer et al., 2016; Fang et al., 2016) based on entity-entity co-occurrence (Nunes et al., 2013) and entity distribution (Aggarwal et al., 2015; Shen et al., 2012b), category embedding is independent of annotated data and can have more training data by collecting multiple entities which share the same category. In KGs, an entity usually has multiple categories from general to specific describing different aspects of entity. For example, entity *dbr:John_Noble* has categories from general to specific (e.g. dbo:Person, yago:Director, dbc:Australian_Film_Actor). Furthermore, a category normally annotates

| dbc:Machine_learning | | | |
|---|---|---|---|
| Category | Similarity | Word | Similarity |
| dbc:Classification_algorithms | 0.774 | algorithm | 0.561 |
| dbc:Artificial_intelligence | 0.761 | logical | 0.548 |
| dbc:Machine_learning_algorithms | 0.758 | computation | 0.545 |
| dbc:Structured_prediction | 0.706 | analysis | 0.540 |
| dbc:Computational_learning_theory | 0.705 | numerical | 0.525 |
| dbc:Learning | 0.683 | parameter | 0.522 |
| dbc:Artificial_neural_networks | 0.667 | method | 0.519 |
| dbc:Computational_complexity_theory | 0.649 | heuristic | 0.505 |
| dbc:Cognitive_science | 0.639 | predictive | 0.498 |
| dbc:Learning_methods | 0.627 | process | 0.497 |
| dbc:Decision_theory | 0.625 | calculation | 0.495 |
| dbc:Algorithms_and_data_structures | 0.623 | unsupervised | 0.491 |
| dbc:Statistical_natural_language_processing | 0.619 | knowledge | 0.486 |
| dbc:Data_mining | 0.604 | learn | 0.479 |
| dbc:Decision_trees | 0.599 | mathematic | 0.476 |

Table 4.9: Examples of top-15 similar categories and words of dbc:Machine_learning

multiple entities indicating their common categorical feature (e.g. *yago:Director* groups all the directors). Because of these characteristics of entities and categories in KG, category vectors would be used more frequently since they are shared by multiple entities, while entity vectors consume more storage space but are used less frequently. Also, combining multiple category vectors to represent an entity vector can capture a more complete meaning of an entity because it combines entity's different aspects. In this way, entity vectors constructed from category vectors may be more effective than those entity vectors from entity embedding. In addition, given that categories are usually constructed hierarchically from general to specific into concept taxonomies, more general categories subsume more specific categories. Correspondingly, entities are annotated with categories from general to specific. In consequence, while training Category2Vec model, more general categories appear more frequently, thus they are related to more various words, whereas more specific categories have less collocation with words. Examples of similar categories and words in trained Category2vec model have been shown in Table 4.8 and Table 4.9, which uses DBpedia abstracts and categories. The category *dbc:Machine_ learning* is sub-category of *dbc:Artificial_ intelligence.* By showing their top 15 similar categories and words, we illustrate that the more general category is more similar to general categories and words, while more specific category is more similar to those specific categories and words. The comparison example shows the property of Category2Vec in capturing the generality and specificity of categories in shared vector space.

In summary, Category2Vec model groups larger numbers of general words into general categories and smaller numbers of specific words into specific categories. In other words, general categories can be viewed as a larger word cluster and specific categories correspond to a smaller word cluster which is contained in the larger word cluster. With this property, Category2Vec model can be used to compare various semantic distance for word-word, word-category, and category-category. In this chapter, we mainly investigate the effectiveness in measuring word-word and word-category similarity for NED.

### 4.2.3 Evaluation

We evaluate the proposed methods with various datasets and answer the research questions through experimental result analysis.

#### 4.2.3.1 Datasets and Implementations

We collected three NEL datasets which are publicly available, including search engine queries, web question answering queries and tweets. As we mainly focus on evaluating the effectiveness of similarity-based NED, we processed the original datasets according to two criteria: (1) the annotated mentions are ambiguous and have more than one candidates in DBpedia; (2) the mention contexts have at least one common noun. The first criteria is used for testing NED specially, while the second criteria is used to create a fair comparison framework for all the similarity methods since knowledge-based semantic similarity methods mostly work for nouns in WordNet. We describe the details of dataset preparation correspondingly for each datasets as below:

*Web Queries* (Hasibi et al., 2015) contains queries derived from *Y-ERD* that offers 2398 entity-annotated queries collecting from the entity recognition and disambiguation challenge (Carmel et al., 2014a) and Yahoo Search Query Log to Entities (Hasibi et al., 2015). We only remain those instances that have been annotated with DBpedia entities resulted in 1151 instances. After filtering based on our criteria, we finally got 340 queries for

| Dataset | #Orignal | #NEL | #NED | #Context | #Candidates | Example |
|---|---|---|---|---|---|---|
| Web Queries | 2398 | 1151 | 340 | 2.2 | 8.4 | "the music man songs " |
| Web Questions | 3778 | 2019 | 587 | 1.9 | 6.7 | "What movies did John Noble play" |
| Tweets | 6025 | 6025 | 2284 | 3.2 | 9.6 | "Five New Apple Retail Stores Opening Around the World." |

Table 4.10: Dataset Statistics

our experiment. Note that since queries are too short, we also include context words from entity mentions (e.g. contextual words music and song are extracted from entity mention "music man song" ).

*Web Questions* (Berant et al., 2013) dataset contains thousands of question answer pairs and each question has been annotated with a Freebase (Bollacker et al., 2008) entity. We have converted those Freebase entities to the corresponding DBpedia entities using a mapping dataset provided by DBpedia (same-as relation). From the original 3778 training dataset, we successfully mapped 2019 entities to DBpedia. After filtering, we finally got 587 questions for our experiment.

*Tweets* (Cano et al., 2016) dataset is the entity linking dataset of Named Entity rEcognition and Linking Challenge (NEEL) at the #Microposts2016, whose task consists of recognizing named entity mentions and their types from English tweets, and linking them to corresponding DBpedia entries with proper entity disambiguation. We use the training dataset containing 6025 annotated English tweets and after filtering we have got 2284 instance for our experiment.

The details of dataset statistics and example text of each dataset are illustrated in Table 4.10. We have shown the average number of contextual words in column #Context and average number of candidate entities in column #Candidates. As expected, the tweet dataset contains more contextual words and ambiguous candidates due to its relatively larger size of text and open domain entities. The web query dataset contains more ambiguous candidates than the question dataset because its entity mentions represent many products such as movies, novels, albums, and those products have different versions. Moreover, in web queries dataset, we have taken into consideration of the mention words as contextual words since they contain meaningful common nouns (e.g. song and novel). In case of question dataset, the factual queries focus on asking questions about people, place and organization, thus the entity mentions are mostly proper nouns, while the ambiguity level of mentions is relatively lower than other two datasets.

All the datasets described above contain original text, annotated mentions, and corresponding gold standard DBpedia entities, whereas the candidate entities are missing. In order to generate candidates for each annotated mention, we created an entity name dictionary using DBpedia datasets of English titles[3] and redirects[4]. The DBpedia titles are actually created from Wikipedia page titles and the redirects are extracted from the redirect pages consisting of a redirection hyperlink from an alternative name to the article indicating synonyms or aliases such as acronyms, common misspellings to name a few. By mapping entity resources in two datasets, we have created an entity name dictionary where various forms of entity names are mapped to a set of DBpedia entities sharing the same lexical names. Then, entity candidates are generated for each dataset by performing exact string matching over annotated entity mentions and entity names in the dictionary. To guarantee the filtering criteria in preparing datasets, those entity mentions having no matching of candidate entities from the entity dictionary, have been removed from the datasets. Thus, all the mentions would have at least two candidate entities. Those entity mentions have and only have a single match of entity are discarded as well in order to focus on testing the performance of NED specially.

We use English abstracts[5] and categories[6] of DBpedia as NED features for different semantic similarity models. To develop IR and LSA models, we use Gensim[7] to process entity abstracts and index them. For those NLP tasks of tokenization, part of speech tagging, we use the spaCY[8], while the lemmatisation is based on NLTK[9]. Moreover, we use Sematch(Zhu and Iglesias, 2016) tool to compute knowledge-based semantic similarity of words using WordNet, while we use the implementation of Word2Vec from Gensim for training word embedding. Category2Vec is built based on the Doc2Vec implementation of Gensim. By joining entity abstracts and categories, we trained category and word embedding using the CBOW model with 300 dimensions.

#### 4.2.3.2 Experimental Settings and Results Analysis

Through the evaluation of NED baseline approaches (TF-IDF and LSA), SCSNED, and Category2Vec in the prepared datasets mentioned above, we want to address the following research questions (RQs):

[3] http://wiki.dbpedia.org/Downloads2015-04#titles

[4] http://wiki.dbpedia.org/Downloads2015-04#redirects

[5] http://wiki.dbpedia.org/Downloads2015-04#extended-abstracts

[6] http://wiki.dbpedia.org/Downloads2015-04#articles-categories

[7] https://radimrehurek.com/gensim/

[8] https://spacy.io/

[9] https://www.nltk.org

| | Web Queries | | | | Web Questions | | | | Tweets | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | Accuracy | Precision | Recall | F-Measure | Accuracy | Precision | Recall | F-Measure | Accuracy | Precision | Recall | F-Measure |
| TF-IDF | 0.421 | 0.582 | 0.421 | 0.459 | 0.496 | 0.523 | 0.496 | 0.503 | 0.338 | 0.671 | 0.338 | 0.397 |
| LSA | 0.459 | 0.626 | 0.459 | 0.500 | 0.579 | 0.607 | 0.579 | 0.588 | 0.399 | 0.698 | 0.399 | 0.470 |
| Word2Vec DBpedia | 0.453 | 0.604 | 0.453 | 0.489 | 0.733 | 0.743 | 0.733 | 0.735 | 0.440 | 0.683 | 0.440 | 0.493 |
| Word2Vec Wikipedia | 0.524 | 0.676 | 0.524 | 0.561 | 0.753 | 0.761 | 0.753 | 0.755 | 0.468 | 0.709 | 0.468 | 0.523 |
| Word2Vec GoogleNews | 0.500 | 0.643 | 0.500 | 0.539 | 0.698 | 0.710 | 0.698 | 0.702 | 0.490 | 0.701 | 0.490 | 0.542 |
| WordNet Path | 0.529 | 0.616 | 0.529 | 0.555 | 0.734 | 0.740 | 0.734 | 0.736 | 0.474 | 0.668 | 0.474 | 0.524 |
| WordNet Wu & Palmer | 0.494 | 0.592 | 0.494 | 0.519 | 0.700 | 0.706 | 0.700 | 0.702 | 0.447 | 0.634 | 0.447 | 0.491 |
| WordNet Resnik | 0.508 | 0.611 | 0.508 | 0.541 | 0.716 | 0.722 | 0.716 | 0.717 | 0.451 | 0.656 | 0.451 | 0.501 |
| WordNet Lin | 0.535 | 0.617 | 0.535 | 0.560 | 0.721 | 0.725 | 0.721 | 0.721 | 0.480 | 0.662 | 0.480 | 0.527 |
| WordNet Jiang & Conrad | 0.538 | 0.604 | 0.538 | 0.562 | 0.717 | 0.723 | 0.717 | 0.719 | 0.500 | 0.699 | 0.500 | 0.548 |
| WordNet WPath | 0.532 | 0.616 | 0.532 | 0.556 | 0.733 | 0.743 | 0.733 | 0.735 | 0.462 | 0.663 | 0.462 | 0.512 |
| Category2Vec Average | 0.529 | 0.738 | 0.529 | 0.581 | 0.600 | 0.626 | 0.600 | 0.606 | 0.360 | 0.687 | 0.360 | 0.430 |
| Category2Vec Max | 0.591 | 0.717 | 0.591 | 0.630 | 0.765 | 0.766 | 0.765 | 0.765 | 0.483 | 0.682 | 0.483 | 0.529 |

Table 4.11: NED in various size of texts using different similarity methods.

- RQ1: How do different NED approaches compare with different type of texts?
- RQ2: How do different word similarity methods compare with the task of NED?
- RQ3: How do different knowledge-based similarity methods compare with the task of NED?
- RQ4: How do different training corpus affect the performance of word embedding on the task of NED?
- RQ5: How do average and max similarity strategy compare with Category2Vec model in the task of NED?

In order to answer the above research questions, we have implemented all the NED approaches described in Section 4.2.2 and evaluated them with three types of datasets. To answer RQ2, we tested SCSNED approach with both corpus-based and knowledge-based word similarity methods. Specifically, all the knowledge-based similarity method are tested respectively using WordNet. For the corpus-based similarity, to answer the RQ4, we collected three word embeddings which are trained from GoogleNews, Wikipedia and DBpedia abstracts respectively. The original word embedding results of Word2Vec (Mikolov et al., 2013b) trained in GoogleNews is used to represent open domain word embedding, while we trained Wikipedia-based word embedding using English Wikipedia with Gensim Word2Vec tool. As Category2Vec is an extension to Word2Vec, we use the word embedding trained in Category2Vec module based on entity abstracts of DBpedia. In addition, we have implemented and evaluated both average and max similarity strategy of NED using Category2Vec model. We use the standard accuracy, precision, recall and F-measure as metrics for NED (Navigli, 2009), and the evaluation results are shown in Table 4.11. We present the conclusion to answer each research question respectively according to the evaluation results reported in the table.

RQ1. Firstly, the SCSNED and Category2Vec approaches have better performance than baseline approaches in all types of datasets, which shows that the fine-grained meaning comparison is more effective than coarse-grained meaning comparison in task of NED. Secondly, as expected, the LSA has been shown better performance than the basic TF-IDF model, thus we draw conclusion that the dimension reduction is effective in solving vocabulary mismatch problem. Thirdly, word-category similarity methods is relatively better than word-word similarity method in shorter texts such as web queries and web questions, while word-word similarity methods are better in relatively longer texts (e.g. tweets). We think this is because the category vectors have more specific meaning than common words in discriminating the entities according to the contextual words. For example, category *dbc:Machine_ learning* and category *dbc:Classification_ algorithms* are more specific than words *machine, learning, classification, algorithm.* Thus, when the context is limited, more specific meaning would play more important role in deciding the correct meaning of the entity. Finally, as we have used a voting strategy (we used top-10 word similarity) in word-word similarity-based NED, with the relatively more contextual words, word-word similarity methods have better performance since entities have more word features than category features.

RQ2. Knowledge-based similarity methods transform the structural knowledge contained in WordNet into similarity scores, while corpus-based Word2Vec transforms statistical knowledge into similarity scores. Through those similarity scores, external resources such as WordNet, and textual corpora such as Wikipedia, GoogleNews are employed to help NED. With WordNet, since the semantic relations between concepts are fixed in the ontology, the effectiveness of knowledge transformation relies on the effectiveness of similarity methods. In comparison, the effectiveness of Word2Vec model depends on the proper textual corpus for training. The experimental results have shown that corpus-based and knowledge-based similarity methods have similar performance in all types of datasets, since we have only considered general domains (e.g. web queries, web questions and tweets). Comparing the best results obtained by the two types of similarity methods, knowledge-based similarity methods have better results in web queries and tweets, while corpus-based similarity methods have better results in web questions. We think that corpus-based similarity methods capture better relatedness in questions (e.g. movie and actor, shown in Table 4.6), while knowledge-based similarity methods represent more information in same domain (movie, novel, or place). Thus, it is better to combine both types of similarity methods in order to contain both relatedness and domain knowledge. Moreover, the knowledge-based similarity methods are based on a specific domain ontology which has limited their application to the scenarios having a well constructed domain ontology. As ontologies are difficult to develop and maintain, corpus-based similarity methods can play important role as the complement of knowledge-based similarity methods. Furthermore, because of simple computational model, corpus-based similarity methods are easy to apply for covering more updated vocabularies.

RQ3. Previous works of semantic similarity evaluations are based on word similarity dataset containing word pairs which are assigned with similarity scale by human subjects (Zhu and Iglesias, 2016). Such evaluation relies on human evaluation over word pairs which may not have the same performance in real applications (Budanitsky and Hirst, 2006). In the task of NED, we are able to compare the effectiveness of different knowledge-based similarity methods in real world application and datasets. In general, from the experimental results, we have found that Resnik (Resnik, 1995) and Wu & Palmer (Wu and Palmer, 1994) methods have relatively lower effectiveness than other four methods in task of NED. Although the other four similarity methods perform differently in different datasets, the Jiang & Conrad (Jiang and Conrath, 1997) method has relatively better performance in queries and tweets, while the Path (Rada et al., 1989) method performs better in web questions. Since WPath (Zhu and Iglesias, 2016) is a tradeoff between the Path (Rada et al., 1989) and pure IC-based methods, it has been shown with consistent performance in all the datasets, which is better than negative cases of IC-based methods and the Path (Rada et al., 1989), but is lower than their positive cases.

RQ4. Word embedding methods are previously evaluated on word level applications such as word similarity, word analogy, part of speech tagging to name a few. Through the NED evaluation, we are able to show the performance of embedding methods in terms of the different corpus for the real application. In general, more training data, the better performance of corpus-based similarity methods. As expected, to train Word2Vec model, more training data (e.g. Wikipedia and GoogleNews) has better performance than less training data (e.g. DBpedia abstract). Since the domain information also influences the training of word embedding, the word embedding trained from DBpedia abstracts and Wikipedia have better performance than GoogleNews in web queries and web questions. In a more general domain, tweets, the general corpus GoogleNews has performed best. Through such comparison, we demonstrate that the different characteristics of training corpus would affect the application of word embeddings in specific domain application such as NED. Furthermore, as Category2Vec is literally an extension of Word2Vec model, it can obtain both category embedding and word embedding from a unified training model. Since both word vectors and category vector are embedded in a shared vector space, Category2Vec would have more applications than word embedding alone. Note that category-category relation can be used to infer entity-entity relation. In conclusion, we have shown that Category2Vec is suitable for training embedding in KGs where both semantic category and textual descriptions are available for entities.

RQ5. We have provided average and maximum strategies in applying the Category2Vec model for NED and evaluated them in all the datasets respectively. The experimental results have shown that the maximum strategy is better than average one in case of NED. We analyze the results empirically by comparing two strategies. For the average strategy, the contextual words and entity categories are generalized via averaging their respective vectors, in order to test the performance of vector combined meaning. The normalized sum average of contextual words and entity categories seem to be too generalized for representing the specific meaning of the context and entity, while the maximum strategy of selecting most similar word-category pairs between context and entity has a better performance. Each entity category represents a specific aspect of the entity, while each contextual word also represents a specific aspect of the context meaning. Finding the most similar pair of contextual word and entity category is actually aligning the context meaning with entity meaning in terms of specific aspects. In this semantic alignment process, individual words or category vectors help to identify the distinctive similarity between context and entity. In consequence, the max strategy is more effective than simple average strategy in representing distinctive word and category meanings. We think that the average strategy has lower performance in representing discriminative features, whereas it may be more suitable in the case that the generalized meaning representation is needed such as detecting the word analogy or representing generative features.

To summarize, we have answered the research questions through the experiment in the task of NED of web queries, web questions and tweets. We have demonstrated: (1) fine-grained word level meaning comparison is better than the baseline of text level meaning comparison (e.g. TF-IDF and LSA) in case of short texts; (2) both knowledge-based and corpus-based similarity methods are effective for NED while it is better to combine both; (3) knowledge-based similarity methods, Path (Rada et al., 1989), Jiang & Conrad (Jiang and Conrath, 1997) and WPath (Zhu and Iglesias, 2016) are generally better than others in case of NED; (4) corpus-based similarity methods depend on proper selection of domain corpus, while Category2Vec is more effective in generating and representing both word and category vectors in a shared vector space with a uniformed embedding model; (5) the Category2Vec model is effective for NED, while the maximum strategy is better than average strategy in case of similarity-based NED.

#### 4.2.3.3 Comparing to The State of the Art

| System | VoteCombine | Category2Vec | LSA | Word2Vec | WordNet | AGDISTIS | AIDA | Babelfy | PBOH | WAT | DoSeR | KEA | ADEL |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F-score | 0.595 | 0.579 | 0.557 | 0.582 | 0.585 | 0.561 | 0.489 | 0.428 | 0.721 | 0.587 | 0.607 | 0.501 | 0.536 |

Table 4.12: Performance comparison of state-of-the-art systems and the proposed methods.

Having evaluated different similarity methods of SCSNED and Category2Vec, we provide an evaluation of proposed NED approaches to the state of the art NED approaches with all the tweet dataset (6025 instances) in NEEL 2016, because its evaluation results are publicly available. We follow the *strong link match* evaluation (Cano et al., 2016) which specially evaluates the disambiguation effectiveness. Note that in the *strong link match* evaluation, a system needs to map the given entity mention to DBpedia entity candidates and select the most suitable entity, thus the final performance is also influenced by the entity name dictionary and name matching apart from the disambiguation algorithm. In the evaluation described in the previous section, we have avoided such influence, while we include this factor in this section in order to compare to the existing approaches. We use the GERBIL (Usbeck et al., 2015) framework to retrieve the evaluation results of the same tweet dataset for those state of the art NEL systems who have separate NED module available. The *D2KB* model in GERBIL framework is chosen because it is the corresponding evaluation mode that is equivalent to *strong link match*. From NEEL 2016 report (Cano et al., 2016), we get the results of challenge's baseline system ADEL (Plu et al., 2016) whose disambiguation approach is based on string similarity and graph-based algorithm PageRank (Hachey et al., 2011). We also get the results of the system KEA (Waitelonis and Sack, 2016) (best reported system in the challenge), whose disambiguation is based on confidence scoring considering string similarity, weighting graph distance, connected component analysis, entity centrality and density.

From GERBIL, we retrieve the evaluation results of AGDISTIS (Usbeck et al., 2014), AIDA (Hoffart et al., 2011), Babelfy (Moro et al., 2014), PBOH (Ganea et al., 2016), WAT (Piccinno and Ferragina, 2014b), and DoSeR (Zwicklbauer et al., 2016), which have separate NED component. AGDISTIS (Usbeck et al., 2014) is based on string similarity and graph-based HITS algorithm, while AIDA (Hoffart et al., 2011) combines entity prior probability, keyphrase-based context similarity and entity coherence. Babelfy (Moro et al., 2014) models entities in a network through its "semantic signature" based on graph random walk algorithm and identifies entity through iterative process in the subgraph. PBOH (Ganea et al., 2016) is a recent pure NED approach that develops probabilistic graphical model using pairwise Markov Random Fields for disambiguation based on statistics from English

Wikipeda corpus considering anchor text. WAT (Piccinno and Ferragina, 2014b) is a redesigned system of TagMe (Ferragina and Scaiella, 2012) and includes graph-based algorithm for ranking entities in entity graph based on entity relatedness, and vote-based algorithm for local disambiguation. DoSeR (Zwicklbauer et al., 2016) is another recent entity disambiguation framework, which combines entity prior probability, entity relatedness and PageRank based disambiguation algorithm. Its entity relatedness is measured based on cosine similarity of entity embedding vectors which is trained from generated entity sequence corpus using Word2Vec (Mikolov et al., 2013b). In comparison, Category2Vec trains category and word embedding. We have discussed and compared entity embedding and category embedding in Sect 4.2.2.3. These state of art systems cover most of disambiguation approaches described in Sect 4.2.1.2.

We include LSA, SCSNED and Category2Vec (max strategy) to compare with existing systems. For SCSNED, we use the two best performing word similarity models in tweet data, which are Word2Vec based on GoogleNews corpus and Jiang & Conrad (Jiang and Conrath, 1997) similarity method based on WordNet (referred as Word2Vec and WordNet respectively for convenience). Furthermore, we use a simple voting-based ensemble approach, called VoteCombine to combine LSA, Word2Vec, WordNet, and Category2Vec. Since each disambiguation approach returns one single best entity, the VoteCombine chooses the majority one from the four approaches. If each disambiguation approach returns a different entity, we use the one from WordNet based SCSNED approach as it achieves the best performance in tweet dataset in the previous evaluation. The F scores of all the NED systems in tweet dataset are shown in Table 4.12. The evaluation results show that all the proposed NED approaches outperforms NEEL challenge baseline and are competitive to the state of the art approaches. The VoteCombine has better performance than most existing systems but PBOH and DoSeR, because those two systems employ more features such as entity prominence, and trained word-entity, entity-entity coherence from labeled dataset. The proposed word-word and word-category similarity methods can be effective feature to complement such systems. Moreover, since SCSNED and Category2Vec do not rely on a labeled dataset, they can be applied to other KGs that have no labeled dataset like Wikipedia. In summary, by comparing to the current state of the art NED system, we have shown that the proposed similarity-based NED approaches are effective and useful. Especially, the studied word similarity methods and Category2Vec model are effective additional features to be complement of the current systems.

## 4.3 Summary

In this chapter, we have presented semantic disambiguation of words and named entities. For WSD, we have presented Synset2Vec which is a neural model of training synset and word vector representation jointly based on WordNet and its sense-annotated dataset. The experimental results show that the model is effective by investigating vector similarity of synset, word, and text for WSD. Regarding to NED task, we have exploited different semantic similarity methods based on various semantic resources, including self-contained KG features (e.g. entity abstract and category), common sense knowledge from ontology (e.g. WordNet) and textual corpora (e.g. GoogleNews and Wikipedia). We proposed a novel NED approach based on word similarity and evaluated both knowledge-based and corpus-based semantic similarity methods in the task of NED. We have demonstrated and identified effectiveness of different semantic similarity methods in a comparative experiment of evaluating unsupervised similarity-based NED with real world datasets of web queries, web questions and tweets. Moreover, we proposed Category2Vec model to learn vector representation of words and categories jointly in the same shared vector space only dependent on a uniformed embedding model and knowledge of KG (e.g. entity abstracts and categories), without relying on labeled dataset. All the similarity methods and models can also be used in other KG-based applications which require to explore similarity between word, category and entities. The main experimental results have shown that semantic similarity methods are more effective than text similarity methods when the contextual information is scarce. The proposed SCSNED and Category2Vec methods are competitive to the state of the art NED approaches, while the semantic similarity features are shown to be effective and can be used as the complement to the existing approaches.

CHAPTER 5

# Semantic Classification and Entity Search

*Similarity captures closeness of concepts and entities respect to their semantic meaning. Matching concepts or entities based on their semantic similarity is different from simple lexical matching of surface forms which is based on boolean matching. Since similarity methods output meaningful similarity scores, we propose to use those numerical similarity score to construct similarity feature vectors for classification, in order to overcome vocabulary mismatch problem in boolean feature representation, and to learn similarity pattern rather than simple occurrence pattern when training the classification model. We demonstrate the application of similarity based classification in a task of concept classification.*

*Moreover, similarity is not only good at semantic matching, but also useful for expansion. Together with similarity-based disambiguation, we show how to query entities from KG with concept expansion, after linking concepts and entities to natural language queries. We introduce a semantic entity search framework which uses similarity for matching semantic resources, disambiguation, and expansion. To develop the working demo of semantic entity search system, we also propose a rule based approach for SPARQL query construction and retrieve entities through execution of SPARQL queries in DBpedia endpoint.*

## 5.1 Similarity-Based Classification

In order to build a concept classification system, the common approaches first extract features for concepts and then use those features to train the classifier in a specific labeled dataset. The most common feature is most frequent collocating words with predefined concepts. One hot representation has been widely adopted to check whether the input text contains the feature words. This approach has the problem of vocabulary mismatch especially for short input texts which may contain no feature words. In such cases, the zero vectors are not informative to train the classifiers. As feature words represent meanings of concepts, by comparing input words with feature words based on their semantic similarity, we are able to capture how close is the meaning between input text and the given concept. Based on this observation, we proposed a similarity-based framework for concept classification, in which concept's features are represented by frequent collocated words while feature vectors are constructed by computing semantic similarity between input words and feature words. We demonstrate the similarity-based classification framework in the Aspect Based Sentiment Analysis (ABSA) (Pontiki et al., 2015, 2016a) task of aspect category classification by proposing both unsupervised model and supervised model.

ABSA is an evaluation task of the SemEval workshop that provides benchmark datasets of reviews and a common evaluation framework. In SemEval 2015 and 2016, the task sentence-level ABSA has defined a subtask so-called Aspect Category Detection, whose aim is to identify every entity E and attribute A pair, towards which an opinion is expressed in the given text (Pontiki et al., 2015). Specifically, given an input sentence such as "The food was delicious", ABSA needs to detect the E and A pair (category=FOOD#QUALITY) for the target word "food" and to estimate its sentiment either positive or negative. The English dataset has been provided for two domains: Laptops and Restaurants. We have chosen the English restaurants domain of the ABSA of SemEval2016 (Pontiki et al., 2016b). In the restaurant domain, SemEval predefines a set of entities SERVICE, RESTAURANT, FOOD, DRINKS, AMBIANCE and LOCATION, which can be viewed as general aspect categories. Our task of aspect category classification consists in assigning a general aspect category to opinion target words. For example, words such as wine, beverage and soda are classified into ontological parent concept DRINKS, while words such as bread, fish, and cheese are classified into FOOD. Note that only entity E (FOOD) is used as general aspect category and the attribute QUALITY is not considered for simplicity.

This task challenges semantic relatedness methods, especially for corpus-based methods. For instance, in restaurant review corpora, those target words such as fish and wine would appear in same surrounding contexts (e.g. "the fish is delicious and the wine is great"). Since

| Category | Frequent Feature Words |
|---|---|
| SERVICE | service, staff, waiter, waitress, wait, manager, delievery |
| RESTAURANT | place, restaurant, spot, pizza, femme, casa, season |
| FOOD | food, pizza, sushi, dish, menu, fish, chicken, meal, salad |
| DRINKS | wine, drink, beer, selection, bottle, martini, glass, margarita |
| AMBIENCE | atmosphere, place, decor, ambience, music, room, garden |
| LOCATION | view, location, neighborhood, city, place, outdoor, avenue |

Table 5.1: Most Frequent Words Co-occur With Each Aspect Category

| Categories | SERVICE | RESTAURANT | FOOD | DRINKS | AMBIENCE | LOCATION |
|---|---|---|---|---|---|---|
| Numbers | 519 | 228 | 2256 | 54 | 597 | 752 |

Table 5.2: Numbers of Sentences in Evaluation for Each Aspect Category

corpus-based methods are based on calculating co-occurrences of terms in a corpus, they can hardly discriminate terms from different categories that are frequently collocated (e.g. fish and wine). In such scenario, knowledge-based methods are useful to include the structural knowledge from domain taxonomy. As illustrated in a fragment of WordNet in Fig. 3.1, lamb, beef, and seafood are sub-concepts of FOOD category, while coffee, tea and milk are sub-concepts of DRINKS category. Intuitively, semantic similarity methods can be used to measure the taxonomical similarity between target words and aspect category in order to classify the target words into correct aspect category. In the following sections, we introduce unsupervised classification methods and supervised classification method respectively.

### 5.1.1 Unsupervised Classification

The most frequent target words of a category are used as features for representing that category. Features of different aspect categories are illustrated in Table 5.1. Formally, we use $A = \{a_1, \ldots, a_n\}$ to denote a set of aspect categories, and $f(a_i)$ to denote the feature words of a category $a_i$. For a feature word $w_k \in f(a_i)$, we use $weight(w_k) = \frac{count(w_k)}{N_{a_i}}$ to denote the weight of the feature word $w_k$. The $N_{a_i} = \sum_{w_k \in f(a_i)} count(w_k)$ denotes the total count of feature words of category $a_i$. The counts of feature words are derived from the annotated datasets from SemEval ABSA (Pontiki et al., 2015, 2016a). We can define

| Method | Accuracy | Precision | Recall | F-measure |
|---|---|---|---|---|
| Path. (Rada et al., 1989) | .793 | .658 | .736 | .680 |
| Leacock-Chod. (Leacock and Chodorow, 1998) | .788 | .656 | .704 | .662 |
| Wu & Palmer. (Wu and Palmer, 1994) | .769 | .630 | .685 | .637 |
| Li (Li et al., 2003) | .783 | .659 | .701 | .667 |
| Resnik. (Resnik, 1995) | .723 | .560 | .679 | .558 |
| Lin. (Lin, 1998) | .731 | .575 | .674 | .567 |
| Jiang & Conrad. (Jiang and Conrath, 1997) | .732 | .606 | .702 | .609 |
| WPath | **.800** | **.664** | **.741** | **.689** |

Table 5.3: Accuracy, Precision, Recall and F-Measure of Aspect Category Classification using different semantic similarity methods.

a simple aspect category classification framework based on the word semantic similarity method defined in Eq.(3.2), in which different semantic similarity methods are used. Given a sequence of new target words $T = \{w_1, \ldots, w_k\}$, we chose the aspect category $\hat{a}$ that maximizes the following similarity function as the correct category of the $T$.

$$\hat{a} = \underset{a_i \in A}{argmax} \max_{w_j \in T} \sum_{w_k \in f(a_i)} sim_{word}(w_j, w_k) * weight(w_k) \tag{5.1}$$

Given an aspect category $a_i$, the formula sums the semantic similarity scores between the target words and the feature words. The highest similarity score of the target word is chosen to represent the similarity score between $T$ and $a_i$. The aspect category with the highest similarity score would be chosen as the correct aspect category.

We use the restaurant review datasets of ABSA in SemEval-2015 and SemEval-2016 (Pontiki et al., 2015, 2016a). Both datasets contain annotated target words and corresponding category. We have converted the specific categories into general categories, and collected a list of target words and category pairs. As a result, we got a dataset containing 4406 tuples in form of target words and category pairs such as (shellfish, FOOD). The numbers of pairs belong to each category are shown in Table 5.2. Since the dataset contains 6 classes, we use multi-class classification metrics accuracy, macro-average of precision, recall and f-measure as the performance metrics to evaluate the semantic similarity methods. We have implemented the semantic similarity based aspect category classification system and evaluated the classification system in the dataset with different semantic similarity methods. The evaluation results are reported in Table 5.4. We have experimented with different $k$ values of WPath and the best $k = 0.9$ is chosen for the WPath method. This k can be treated as the optimized setting for WPath method in calculating semantic similarity between concepts in the restaurant domain. The $k$ value can provide insight about which metrics perform better in a given group of concepts. Since $k$ has shown higher value in this restaurant domain, the structure information of concept taxonomy is relatively more important. It is also shown in Table 5.4 that the structure based semantic similarity methods, path, lch, wup, and li are performing better than IC based methods res, lin and jcn. Moreover, the WPath method has achieved the best accuracy, precision, recall and F-measure score among other semantic similarity methods.

### 5.1.2 Supervised Classification

The baseline of supervised aspect category classification provided by SemEval employs a Support Vector Machine (SVM) with a linear kernel. Specifically, $n$ unigram features are extracted from the training data, where the category value (e.g., FOOD#QUALITY) of the tuple is used as the correct label of the feature vector (Pontiki et al., 2016b). For each test sentence $s$, a feature vector is built and the trained SVM is used to predict the correct category. This unigram feature representation lacks of the ability in addressing those feature words that are not encountered in the training process. As reported in SemEval (Pontiki et al., 2016b), word clusters learned from Yelp data are used to expand the features. However, those similar words of word clusters are added to feature vectors considering the same weight as the unigram features appearing in the training data, without concerning the different semantic distance between words.

With such concerns, we aim at combining knowledge (e.g. WordNet) and corpus (e.g. Yelp) sources in order to improve aspect classification. Our main contribution is the hybrid model that consists of a word embeddings model (Mikolov et al., 2013a) and semantic similarity model using WordNet (Mihalcea et al., 2006). We propose to use similarity score as the weight of each vector dimension so that the semantic similarity between words computed by word2vec and semantic similarity measures are included for training. Specifically, we explicitly use the $n$ unigrams as feature vector, in which the word similarity between target words and feature words are used to represent each dimension of feature vector. The idea is to train a semantic predictive model for each category based on the feature words and similarity models using SVM. Formally, let $F = \{f_1, f_2, \ldots, f_n\}$ be the set of feature words, a feature vector is represented as $V \in [0,1]^N$. For a set of target words $T = \{w_1, \ldots, w_m\}$, the value of a dimension $f_i$ is computed from $\max_{w_j \in T} sim(w_j, f_i)$, where the $sim$ function denotes the word similarity between two words. The calculation of similarity scores is more computational intensive than counting the occurrence of words. Since the target words are in the form of short text (several words), and the feature vector can be composed by most representative words (small vector dimensions), the intensive computation problem can be alleviated using word similarity matrix.

The *sim* function is implemented by word2vec (Mikolov et al., 2013a) for training Yelp data and the semantic similarity measures based on WordNet (Mihalcea et al., 2006). For word2vec, we have obtained a continuous representation of words, where words that co-occur frequently are mapped to vectors close in vector space. Based on the distributional semantics hypothesis, the words co-occur in a same surrounding context are treated as relevant so that they have high similarity. Consequently, the $sim(w_j, f_i)$ function is implemented as cosine similarity between two word vectors. Using this word2vec similarity model, a first feature vector $V_{word2vec} \in [0, 1]^N$ is obtained.

The word2vec model considers the co-occurrence information of the same surrounding context, which would make a wide variety of words to be considered as related. This would challenge the word2vec model when discriminating words from different categories that are frequently collocated (e.g. food and drink). For instance, in restaurant domain, those target words such as fish and wine would appear in same surrounding contexts (e.g. “the fish is delicious and the wine is great”). If a word2vec model is trained from such corpus simply based on calculating co-occurrences of words, many words belonging to different categories would have similar similarity. In order to solve this problem, semantic similarity methods using WordNet (Mihalcea et al., 2006) are useful to complement the word2vec model by including the structural knowledge from taxonomy. As illustrated in a fragment of WordNet in Fig. 3.1, lamb, beef, and seafood are sub-concepts of FOOD category, while coffee, tea and milk are sub-concepts of DRINKS category. Although WordNet based similarity model can retain taxonomical information from WordNet, it can only address limited words that are contained in WordNet. Combining word2vec similarity model and WordNet similarity model can enable the aspect classification model to have good ability in addressing large vocabularies and encoding hierarchical knowledge of common words from WordNet. In consequence, apart from Word2Vec, we also consider the semantic similarity methods using WordNet.

The semantic similarity methods exploit the hierarchical classification of all words via is-a relation, whose intuition is that two words are more similar if they are closer to each other in WordNet taxonomy. To implement the WordNet based *sim* function, we use all those knowledge-based semantic similarity methods including Path (Rada et al., 1989) , Leacock-Chod (Leacock and Chodorow, 1998) method, the Wu & Palmer (Wu and Palmer, 1994) method, the Resnik (Resnik, 1995) method, Lin (Lin, 1998) method and Jiang &

Conrad (Jiang and Conrath, 1997) method.

A list of feature words are extracted from training data. Apart from the word2vec based feature vector $V_{word2vec}$ mentioned previously, another feature vector $V_{wordnet} \in [0,1]^N$ is composed by computing the semantic similarity between target words and feature words using the WordNet based semantic similarity methods. Consequently, a $2N$ dimension vector is composed for training and classifying new sentences by considering both word2vec similarity model and WordNet similarity model. The evaluation results show that combining word embedding and semantic similarity measures can improve the performance of aspect category classification. We use the SemEval16 dataset of English Restaurant domain dataset. The training dataset consists of 1880 tuples and the test dataset consists of 650 tuples. We extracted most common 10 words of each category and composed into 76 feature words by removing duplicates. The small feature number is not a problem since the vocabularies are contained in word2ve and WordNet. Nevertheless, the quality of feature words should be considered because we use the word similarity scores as the value of feature vectors. We use the most frequent words for simplicity in this article. The word2vec similarity model and WordNet similarity model are used to compute word similarity between target words and feature words. We trained the aspect classification model using the linear kernel of SVM using the sklearn[1] package. The classification metrics accuracy, precision, recall and F-score are used as the performance metrics to evaluate the different models.

We have experimented the classification model in different settings: simple features, knowledge-based features, dense vector features and combined features. The experimental results are shown in Table 5.4. In the simple features, we use the simple word list features $V_{wordlist} \in \{0,1\}^N$, where the word list is the 76 feature words. In this setting, we use the unigram occurrence feature to train a classification module using SVM, and use this model as baseline. Note that the different learning softwares and settings would influence the experimental results so that we implemented a simple baseline following the description of SemEval. In order to show that the similarity based features are more effective than the simple word occurrence features, we extended the simple feature model to the knowledge-based model and dense vector model. In the knowledge-based setting, we have trained and evaluated the classification model using the WordNet based similarity measures respectively. Table 5.4 shows that the Path (Rada et al., 1989) similarity measure is the best metric for aspect classification, and the most of similarity measures are more effective than the baseline except for the Resnik (Resnik, 1995) method. In the dense vector setting, we have used word2vec embedding to learn the word vectors from Yelp comments data, and trained the aspect classification model only with the word2vec similarity model. The experimen-

[1] http://scikit-learn.org

| Method | Corpus & KB | Accuracy | Precision | Recall | F-measure |
|---|---|---|---|---|---|
| Simple Feature | Word List | .745 | .72 | .74 | .71 |
| **Knowledge-based** | | | | | |
| Path. (Rada et al., 1989) | WordNet | .78 | .77 | .78 | .75 |
| Leacock-Chod. (Leacock and Chodorow, 1998) | WordNet | .757 | .73 | .76 | .73 |
| Wu & Palmer. (Wu and Palmer, 1994) | WordNet | .751 | .70 | .75 | 72 |
| Resnik. (Resnik, 1995) | WordNet | .646 | .65 | .65 | .63 |
| Lin. (Lin, 1998) | WordNet | .774 | .73 | .77 | .74 |
| Jiang & Conrad. (Jiang and Conrath, 1997) | WordNet | .768 | .77 | .77 | .74 |
| **Dense Vectors** | | | | | |
| Word2Vec. (Mikolov et al., 2013a) | Yelp | .818 | .79 | .82 | .78 |
| **Combination** | | | | | |
| Word2Vec + Path | WordNet + Yelp | .82 | .80 | .82 | .79 |
| Word2Vec + Leacock-Chod | WordNet + Yelp | .81 | .80 | .81 | .78 |
| Word2Vec + Wu & Palmer | WordNet + Yelp | .813 | .80 | .81 | .78 |
| Word2Vec + Resnik | WordNet + Yelp | .814 | .80 | .81 | .78 |
| Word2Vec + Lin | WordNet + Yelp | .813 | .80 | .81 | .78 |
| Word2Vec + Jiang & Conrad. | WordNet + Yelp | .82 | .80 | .82 | .79 |

Table 5.4: Accuracy, Precision, Recall and F-Measure of Aspect Category Classification using different methods.

tal result shows that the word2vec similarity model is more effective than knowledge-based methods and baseline. By looking at each category, we found that the knowledge-based features are more effective for food and drink categories while word2vec performs better in other categories. Since word2vec feature is trained from a domain corpus (Yelp comments), it has better coverage in vocabularies and the categories such as AMBIENCE, LOCATION are more concerned with relevant features rather than hierarchical feature. In the combined setting, we use both word2vec similarity model and WordNet similarity model to train and evaluate in order to select the best combination between word embedding and semantic similarity methods. Table 5.4 shows that both Path (Rada et al., 1989) and Jiang & Conrad (Jiang and Conrath, 1997) are the best in combining with word2vec, in terms of F measure (.79).

In summary, from the experimental results, we found that the similarity based features are effective in learning the aspect classification model. Furthermore, combining the word embedding model and semantic similarity measure is promising in training aspect classification model, since it has achieved best performance in our experiments, and it can combine the word coocurrence information together with hiearchical knowledge from WordNet.

## 5.2 Semantic Entity Search

As an increasing amount of the knowledge graph is published as Linked Open Data, semantic entity search is required to develop new applications. However, the use of structured query languages such as SPARQL is challenging for non-skilled users who need to master the query language as well as acquiring knowledge of the underlying ontology of Linked Data knowledge bases. In this section, we propose the Sematch framework for entity search in the knowledge graph that combines natural language query processing, entity linking, entity type linking and semantic similarity based query expansion. The system has been validated in a dataset and a prototype has been developed that translates natural language queries into SPARQL.

Entity-centric queries constitute a large fraction of web search queries, and KGs enable the entity-centric information access more convenient. Most of entity-centric queries are in the form of an entity mention together with some contextual words indicating a reference to another entity (capital of Spain) or an intent of the entity's property (population of Spain). Recognition of entity mention and surrounding contexts can improve the search performance such as returning answers directly. Entities are usually not searched alone, but often combined with other semantic information such as types, attributes or properties, relationships or keywords (Pound et al., 2010b). Increasing amounts of structured data are published as Linked Open Data (LOD) in the form of Resource Description Framework (RDF). The KG such as DBpedia (dbp, 2007) and YAGO2 (Hoffart et al., 2013) are examples that have succeeded in creating large general purpose RDF knowledge graphs on the Web of Data, whose knowledge is extracted from Wikipedia. Those initiatives have enabled the KG to change the web from a web of documents into a web of entities. Hence, apart from identifying a single entity based on its textual description, retrieving a list of entities from KG conforming user's specific information needs is also important for both web users and web applications. For example, when a student wants to compare universities in Spain or a web application needs to display all the universities in Spain, both cases require a list of entities of type University with the restriction of Location Spain.

However, querying a list of entities from these heterogeneous structured KGs is challenging for non-skilled users who need to master the syntax of a structured query language (such as SPARQL) and to acquire sufficient knowledge of the underlying ontology (schema and vocabulary). The ideal way for casual users to query from KGs is using Natural Language Interfaces (NLI), where users can express their information needs using Natural Language (NL) without being aware of the heterogeneous LOD vocabulary. The research in NLI for KGs has its roots in the application of traditional keyword-based information retrieval techniques to indexed RDF data such as the works in semantic search (Tummarello et al., 2007; Cheng and Qu, 2009). Recent researches such as (Zhou et al., 2007; Shekarpour et al., 2011; Freitas et al., 2011; Lopez et al., 2012; Damljanovic et al., 2012; Unger et al., 2012; Shekarpour et al., 2014) have focused on advanced Question Answering (QA) techniques over KGs by translating NL queries into formal SPARQL queries. In this chapter, we have restricted the queries to queries with just one relation, called Single Relation Type-based Queries (SRTQs) such as full sentence query *Give me all the universities located in Spain.* An abbreviated version of SRTQ can be expressed with keywords, i.e. *universities Spain.* This example of SRTQ can be rewritten as an equivalent conjunctive formal logic expression $?x \leftarrow (?x, is, University) \cap (?x, ?relation, Spain)$ where ontology class *University*, and instance *Spain* are restrictions on the variable $x$.

To clarify the task of semantic entity search for SRTQ, we give the formal definitions as follows. A Knowledge Graph $K$ is a directed graph $G_k = \langle C, I, R, L, \tau \rangle$ (Zhou et al., 2007), where $C$ and $I$ define the sets of *class* and *instance*; $R$ and $L$ are the sets of *relation* and *literal*; and $\tau$ is a function $(C \cup I) \times (C \cup I \cup L) \rightarrow R$ that defines all triples in $K$. Let $Q$ a SRTQ expressed in NL. $Q = (q_1, q_c, q_i..q_n)$ is a bag of terms containing entity type mention $q_c$ and entity instance mention $q_i$. Entity Linking is defined as $f_e : q_i \rightarrow e \in I$ and Type Linking is defined as $f_t : q_c \rightarrow t \in C$. The formal query $F : \langle e, t, \tau' \rangle$ over $K$ is a graph $G_f$ subsumed by $G_k$. From the definitions above, the entity search task for SRTQ can be modeled as: given $Q$, detect and link entity type $t$ and entity instance $e$ to K via $f_e$ and $f_t$, constructing and executing formal queries $\{F\}$ over $K$ to get desired entities.

For example, in the query described above *query(Spain, university)*, the results of this query are the entities whose entity type is *University* and have semantic relatedness (*located-in*) with the mentioned entity instance *Spain*. By linking *university* and *Spain* to their proper URIs in $K$, the formal query $< Spain, university, ?relation >$ can be translated into SPARQL query. By executing this query in a specific SPARQL endpoint, a list of university entities can be retrieved from a specific KG. Note that the relation terms such as *located-in* in the user query is not detected and mapped to $R$. The relation is used as a variable ($?relation$) in the query construction. In the current work, both the desired entities and the corresponding relation with the mentioned entity are returned as search results, where the relations are implemented as facets for faceted browsing for end users. One of our future works is to include relation information for improving the search performance.

In this section, we propose a framework for semantic entity search in SRTQ over heterogeneous KGs. Since both the entity types mentioned in a user query and the ontology classes for annotating entities in KG (rdf:type) may be too general or too specific, a semantic similarity based type expansion algorithm is proposed and implemented for ontology class enrichment in SPARQL query construction in order to bridge this vocabulary gap. A dataset for SRTQ has been collected to evaluate both the Sematch framework and the proposed algorithm. including a working demo using DBpedia SPARQL endpoint.

### 5.2.1 Related Works

Several NLI systems have been developed for keyword-based search or QA over KG. Semantic keyword-based search system Sindice (Tummarello et al., 2007) is an adaptation of conventional document retrieval approach for RDF data. Keyword-based entity search system Falcons (Cheng and Qu, 2009) relies on matching query keywords in indexed terms. SPARK (Zhou et al., 2007) translates keyword queries into formal logic queries to facilitate end users to perform semantic search. Treo (Freitas et al., 2011) combined entity search, semantic relatedness and spreading activation search to query over LOD using NL queries. PowerAqua (Lopez et al., 2012) is an ontology-based QA system which can combine information from heterogeneous LOD. FREyA (Damljanovic et al., 2012) uses syntactic parsing in combination with the ontology-based lookup, as well as user interaction in order to interpret the question. Unger et al. (Unger et al., 2012) presented a QA system relying on deep linguistic analysis in generating SPARQL templates for answering more complex questions. SINA (Shekarpour et al., 2014) is a keyword search system that can perform QA tasks by transforming keywords or NL queries into conjunctive SPARQL queries over LOD sources.

Sematch is a keyword-based entity search system especially for answering SRTQs aiming to retrieve a list of entities. It followed the approach (Shekarpour et al., 2011) in which SPARQL queries are constructed from mapping keywords to LOD URIs and filling URIs into predefined graph patterns. Sematch adopted the idea of using WordNet taxonomy for interlinking entity type vocabulary like the work (Ballatore et al., 2014) and proposed semantic similarity based type expansion algorithm for enriching type information in generating SPARQL queries. Query expansion for LOD has also been proposed in (Augenstein et al., 2013) and (Shekarpour et al., 2013). Augenstein et al. (Augenstein et al., 2013) mainly

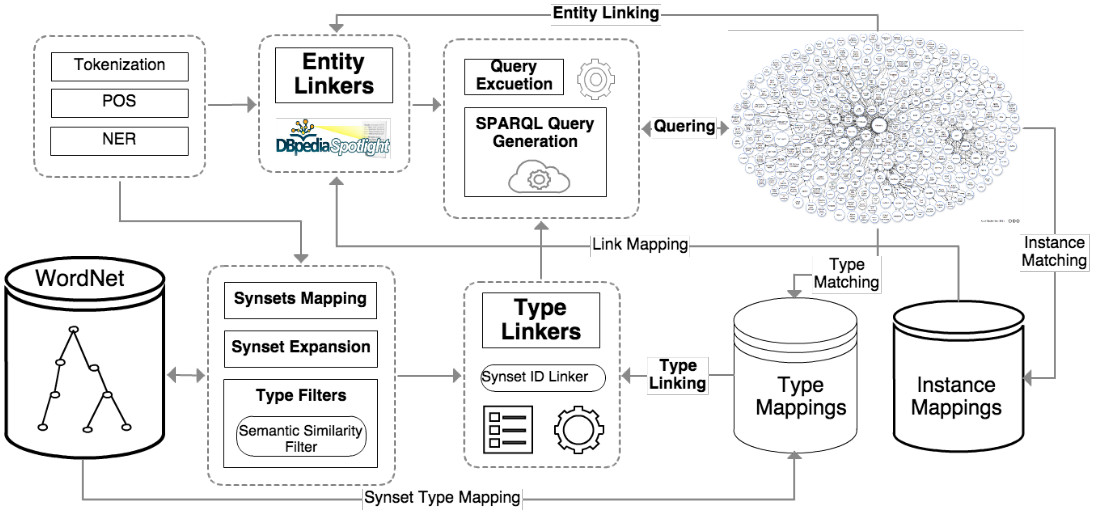


Figure 5.1: Entity Search Framework Overview

focused on mapping keywords to LOD and relying on KG for query expansion. Shekarpour et al. (Shekarpour et al., 2013) used machine learning approaches to combine expansion features from both WordNet and LOD and applied them in semantic search. Sematch focused on expanding entity types with WordNet hypernyms/hyponyms and using semantic similarity measures to optimize precision.

### 5.2.2 Entity Search Framework

The overall architecture of Sematch framework is shown in Figure.5.1. The NL query processing component performs NLP tasks of tokenization, Part of Speech Tagging and Name Entity Recognition (NER) using NLTK[2]. Then, the entity linking component detects the named entity and maps it to instance URI of the KG. In the type expansion component, the type mentioned in the query is mapped to WordNet synsets and expanded based on WordNet taxonomy. Then, type synsets are mapped to ontology class URIs of the KG through *Synset ID Linkers*. Finally, SPARQL queries are generated based on the type and entity URIs obtained before in the *Query Engine*. In this section, we describe the details of entity linking, type expansion and the query graph generation.

The entity linking (Rao et al., 2013) component takes all the tokens except for stopwords. Those tokens are required because the task of entity linking not only links entity mentions that occur in query tokens to entries in the KG but also discriminate entity mentions. Nevertheless, only the links of entities (Location, Person, etc.) recognized by the NER will

[2] http://www.nltk.org/

be sent to the query construction engine. In the example query described above, the entity mention *Spain* is detected and mapped to URI *DBpedia:Spain*. We use NER together with our NED for entity linking. The other linking service such as DBpedia Spotlight (Mendes et al., 2011) can also be used conveniently.

We then introduce the details of translating $q_c$ into entity type $t$. The query $q_c$ is first mapped to a list of WordNet (Miller, 1995) synsets based on their specific sense in the query through WSD module introduced in previous chapter. Unlike conventional IR using synsets for synonym expansion, *synsets mapping* reconciles words to synsets with specific meaning. Thus, the types for describing things are processed at the semantic level (meanings) rather than at the lexical level (terms). WordNet provides relations between synsets such as hypernymy/hyponymy (i.e., the relation between a sub-concept and a super-concept) and holonymy/meronymy (i.e., the relation between a part and the whole). The synset type seeds from *synsets mapping* are expanded based on WordNet hypernyms/hyponyms. Though the recall can be increased by expanding with hypernyms/hyponyms, it is also important to guarantee a certain level of precision. Since semantic similarity measures the proximity between synsets mainly based on hierarchical relation (Is-A), semantic similarity is applied in type expansion for optimizing its precision. Let $\Sigma_{synset}$ be all the noun synsets in WordNet. The *semantic similarity function sim* : $\Sigma_{synset} \times \Sigma_{synset} \rightarrow [0, 1]$ is defined as a list of the state of art semantic similarity measures including edge counting based measures *path* (Rada et al., 1989), *wup* (Wu and Palmer, 1994), *lch* (Leacock and Chodorow, 1998), and information content based measures *res* (Resnik, 1999), *jcn* (Jiang and Conrath, 1997), *lin* (Lin, 1998). A threshold $\eta \in [0, 1]$ is used to establish the semantic similarity between two synsets: $sim\,(s_1, s_2) >= \eta$. Let $\Sigma_{seeds}$ denote the synset type seeds from *synsets mapping* component, the semantic similarity based type expansion algorithm is defined in Algorithm 3. The final algorithm returns a list of expanded synsets which are also merged into a synset type list.

A synset type list is a set of synsets including seed synsets and expanded synsets. Before constructing the query, expanded synsets have to be transformed into proper URIs with *Synset ID Linkers*. A *Synset ID Linker* is an implementation of the Type Linking function $f_t : q_c \rightarrow t \in C$, which links synsets to the Linked Data ontology classes by looking up the type mapping data[3]. The type mapping data[4] is derived from *yagoDBpediaClasses* and *yagoWordnetIds* in YAGO2. In this form, URIs of ontology classes from different knowledge graphs are unified by WordNet synsets based on their meanings. Some DBpedia ontology[5]

[3] university.n.01, http://dbpedia.org/class/yago/University108286163

[4] Mapping Data contains 68423 entries of synsets and YAGO ontology classes.

[5] 145 DBpedia ontology classes are aligned to the mapping data.

**Algorithm 3** Semantic Similarity Based Synset Expansion

```
procedure EXPANSION($\Sigma_{seeds}, \eta, sim$)
    $\Sigma_{result} \leftarrow \emptyset$
    for all $s \in \Sigma_{seeds}$ do
        EXPAND($s, s, \eta, sim, \Sigma_{result}$)
    end for
    return $\Sigma_{result}$
end procedure
procedure EXPAND($c, s, \eta, sim, \Sigma$)
    $\Sigma \leftarrow c$
    for all $x \in hypernyms(c)$ do
        if $x \notin \Sigma$ and $sim\,(s, x) >= \eta$ then
            EXPAND($x, s, \eta, sim, \Sigma$)
        end if
    end for
    for all $y \in hyponyms(c)$ do
        if $y \notin \Sigma$ and $sim\,(s, y) >= \eta$ then
            EXPAND($y, s, \eta, sim, \Sigma$)
        end if
    end for
end procedure
```

classes are aligned to the type mapping data based on the data[6] provided by YAGO2. Ontology classes in other knowledge graphs can also be aligned to WordNet synsets based on the current type mapping data using ontology alignment techniques (Ballatore et al., 2014). After type expansion, the entity mention *university* is expanded into a list of ontology class URIs. In the following section, we describe how to construct the formal query $F$ using $e$ and $t$ based on predefined graph patterns.

### 5.2.3 SPARQL Query Generation

Given URIs of $e$ and $t$, SPARQL queries can be constructed using Graph Pattern Collection (GPC) for SRTQ derived from the graph patterns defined in (Shekarpour et al., 2011). GPC is a set of triple patterns and is defined as: $GPC = \big\{(s, p, o) | (s \in I \vee s = variable) \wedge (p = variable) \wedge (o \in I \vee o \in C \vee o = variable)\big\}$. The Graph Pattern

[6] http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/linking/

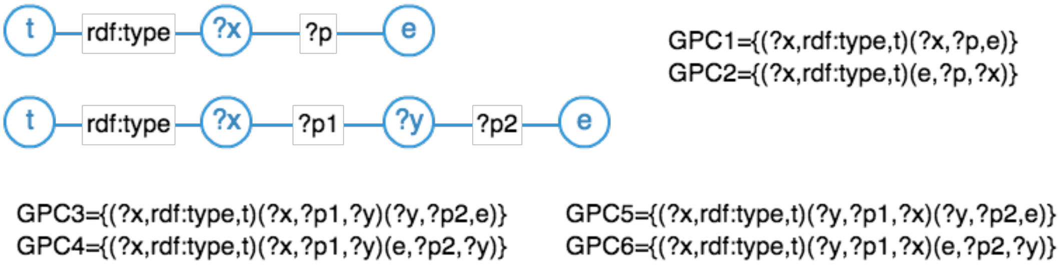

t
rdf:type
?x
?p
e
GPC1={(?x,rdf:type,t)(?x,?p,e)}
GPC2={(?x,rdf:type,t)(e,?p,?x)}
t
rdf:type
?x
?p1
?y
?p2
e
GPC3={(?x,rdf:type,t)(?x,?p1,?y)(?y,?p2,e)}
GPC4={(?x,rdf:type,t)(?x,?p1,?y)(e,?p2,?y)}
GPC5={(?x,rdf:type,t)(?y,?p1,?x)(?y,?p2,e)}
GPC6={(?x,rdf:type,t)(?y,?p1,?x)(e,?p2,?y)}

Figure 5.2: Graph Pattern Collections

**Algorithm 4** Query Generation and Execution

```
procedure ENGINE( $t$, $e$, $GPS$, $G_k$)
    $\Sigma_{result} \leftarrow \emptyset$
    $T \leftarrow Union(t)$
    for all $GPC \in GPS$ do
        $F \leftarrow construct(GPC, T, e)$
        $\Sigma_{result} \leftarrow query(F)$
    end for
    return $HashSet(\Sigma_{result})$
end procedure
```

Set (GPS) is a set of all GPCs and is represented as $GPS = \{g|g = GPC\}$ which are $\{GPC_1, GPC_2, GPC_3, GPC_4, GPC_5, GPC_6\}$. The details of the graph patterns for each GPC are illustrated in Figure.5.2. In these pattern collections, symbols preceded by question marks denote variables and symbols without question marks are $t$ (entity type) and $e$ (entity instance).

Those patterns are only valid for certain combinations with $t$. The goal of type expansion is to generate adequate type URIs. The *Union* syntax of SPARQL query language is used to combine all the available type URIs such as (?x, rdf:type, t1) Union (?x, rdf:type, t2). $GPC_1$ and $GPC_2$ represent direct semantic relation with the mentioned entity, which is shown in the first pattern graph of Figure. 5.2. Semantic relation expansion is represented by $\{GPC_3, GPC_4\}$ and $\{GPC_5, GPC_6\}$. The relation expansion is included because the relations between entities in the KG can be transitive relations. Finally, $t$ and $e$ are constructed into $F$ by being filled into all GPCs. The queries are sent to the user specified SPARQL endpoint and the results are unified by removing repetitions. The query construction and execution process are illustrated in Algorithm 4. The example of $GPC_1$ for constructing the query *university Spain* is illustrated as below:

```
SELECT DISTINCT ?x ?p WHERE {
{ ?x rdf:type dbpedia:University> } UNION
{ ?x rdf:type yago:University108286163 } UNION
{ ?x rdf:type yago:CityUniversity103036244 } UNION
{ ?x rdf:type dbpedia:EducationalInstitution> } UNION
{ ?x rdf:type yago:EducationalInstitution108276342 } .
?x ?p <http://dbpedia.org/resource/Spain> .
} GROUP BY ?x
```

### 5.2.4 Evaluation

In this section, we evaluate the performance of entity search framework. The evaluation aims to achieve three goals: 1) compare the effectiveness of different semantic similarity methods for type expansion 2) evaluate the feasibility of semantic similarity based type expansion; 3) compare the effectiveness of relation expansion by using different numbers of GPCs.

We have collected a dataset for SRTQs from a dataset for entity search in DBpedia (Balog and Neumayer, 2013) which contained data from several campaigns, including INEX-XER, TREC Entity, SemSearch ES, SemSearch LS, QALD-2, and INEX-LD. Table.5.5 illustrates our 29 SRTQs. For convenience, we have also shown the queries with detected entity type mention and entity instance mention. Precision and recall were used as our metrics. Assuming A is the relevant set of entities for the query that is provided in dataset, and B is the set of retrieved entities by running Sematch, the precision and recall can be defined as follows:

$$Recall = \frac{|A \cap B|}{|A|} \tag{5.2}$$

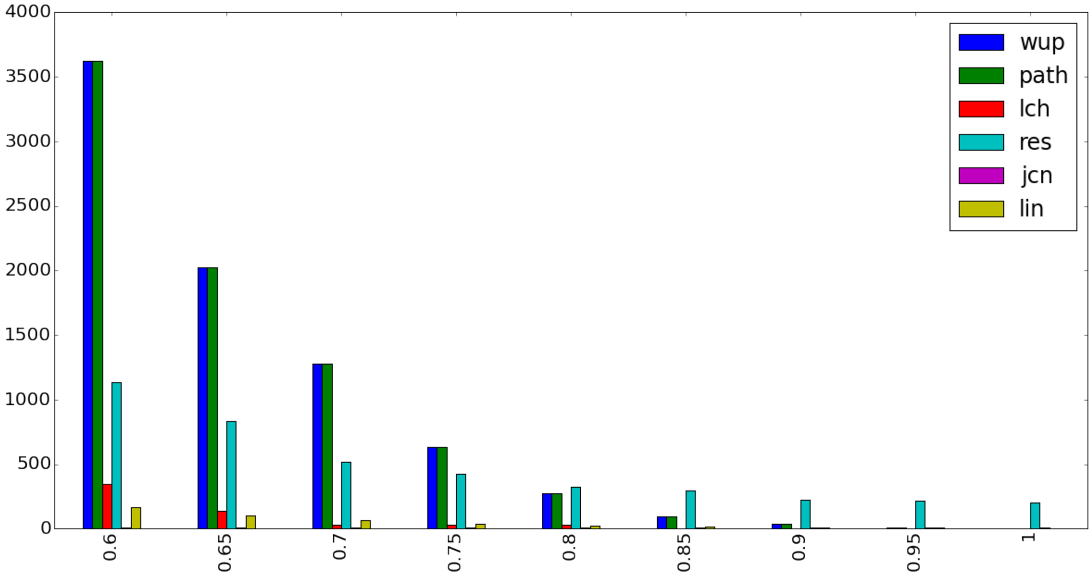
$$Precision = \frac{|A \cap B|}{|B|} \tag{5.3}$$

where $|.|$ gives the size of the set and $|A \cap B|$ is the set of entities that are both relevant and retrieved. Fig.5.3 illustrates the counts of expanded synsets using different semantic similarity methods as threshold varying from 0.6 to 1 with interval of 0.05. The semantic similarity methods *wup* and *path* have the same performance in expanding synsets so we only compare the method of *wup*, *lch*, *res*, *jcn*, and *lin*. In order to limit the maximum number of expanded synsets under 50, the thresholds of 0.9, 1.0 are chosen where 1.0 represents the baseline without expansion and 0.9 represents the type expansion. Furthermore, we use two sets of GPCs for comparing which are $gp1 = \{GPC_1, GPC_2\}$ and $gp2 = \{GPC_1, GPC_2, GPC_3, GPC_4\}$. The direct relation between desired entity and mentioned entity is represented by $gp1$, while

| ID | Source | Query | Type | Entity |
|---|---|---|---|---|
| 1 | INEX_LD-20120131 | vietnam travel national park | park | dbpedia:Vietnam |
| 2 | INEX_LD-20120132 | vietnam travel airports | airports | dbpedia:Vietnam |
| 3 | INEX_LD-2010004 | Indian food | food | dbpedia:India |
| 4 | INEX_XER-62 | Neil Gaiman novels | novels | dbpedia:Neil_Gaiman |
| 5 | INEX_XER-72 | films shot in Venice | film | dbpedia:Venice |
| 6 | INEX_XER-79 | Works by Charles Rennie Mackintosh | works | dbpedia:Charles_Rennie_Mackintosh |
| 7 | INEX_XER-86 | List of countries in World War Two | countries | dbpedia:World_War_II |
| 8 | INEX_XER-91 | Paul Auster novels | novels | dbpedia:Paul_Auster |
| 9 | INEX_XER-108 | State capitals of the United States of America | capitals | dbpedia:United_States |
| 10 | INEX_XER-124 | Novels that won the Booker Prize | novels | dbpedia:Man_Booker_Prize |
| 11 | INEX_XER-125 | countries which have won the FIFA world cup | countries | dbpedia:FIFA_World_Cup |
| 12 | INEX_XER-133 | EU countries | countries | dbpedia:European_Union |
| 13 | INEX_XER-139 | Films directed by Akira Kurosawa | film | dbpedia:Akira_Kurosawa |
| 14 | INEX_XER-140 | Airports in Germany | airports | dbpedia:Germany |
| 15 | INEX_XER-141 | Universities in Catalunya | university | dbpedia:Catalonia |
| 16 | QALD2_te-6 | Give me all professional skateboarders from Sweden | skateboarders | dbpedia:Sweden |
| 17 | QALD2_te-17 | Give me all cars that are produced in Germany | car | dbpedia:Germany |
| 18 | QALD2_te-28 | Give me all movies directed by Francis Ford Coppola | movie | dbpedia:Francis_Ford_Coppola |
| 19 | QALD2_te-39 | Give me all companies in Munich | companies | dbpedia:Munich |
| 20 | QALD2_te-60 | Give me a list of all lakes in Denmark | lakes | dbpedia:Denmark |
| 21 | QALD2_te-63 | Give me all Argentine films | film | dbpedia:Argentina |
| 22 | QALD2_te-82 | Give me a list of all American inventions | invention | dbpedia:United_States |
| 23 | QALD2_tr-16 | Give me the capitals of all countries in Africa | capitals | dbpedia:Africa |
| 24 | QALD2_tr-53 | Give me all presidents of the United States | presidents | dbpedia:United_States |
| 25 | QALD2_tr-63 | Give me all actors starring in Batman Begins | actors | dbpedia:Batman_Begins |
| 26 | QALD2_tr-68 | Which actors were born in Germany? | actors | dbpedia:Germany |
| 27 | QALD2_tr-70 | Give me all films produced by Hal Roach | film | dbpedia:Hal_Roach |
| 28 | QALD2_tr-78 | Give me all books written by Danielle Steel | book | dbpedia:Danielle_Steel |
| 29 | QALD2_tr-84 | Give me all movies with Tom Cruise | movies | dbpedia:Tom_Cruise |

Table 5.5: The Query Dataset Used in Evaluation.

Figure 5.3: Synset Expanding based on Thresholds



*gp*2 represents relation expansion. We use the DBpedia SPARQL endpoint[7] to execute SPARQL queries. The experiment results are shown in the following section.

Within the experimental configuration defined in the previous subsections, each query in Table 5.5 has been executed 20 times with two thresholds (th=0.9 and th=1.0), two sets of GPCs (gp1 and gp2), and five semantic similarity measures. However, among those queries, the current prototype of Sematch is unable to answer the queries 5, 6, 8, 11, 22, 23 and 28. Thus, we have collected the results of 76% queries in the evaluation dataset. For each of those queries, 20 precision and recall values are collected. The average of those values have been illustrated in Table 5.6 with the corresponding settings. Each column of this table represents the specific semantic similarity measures which are *wup* (Wu and Palmer, 1994), *lch* (Leacock and Chodorow, 1998), *res* (Resnik, 1999), *jcn* (Jiang and Conrath, 1997) and *lin* (Lin, 1998). Each row of the table represents the specific settings of threshold and GPCs. For each cell, the average precision and recall are presented as (precision, recall) correspondingly.

The results have shown that the Sematch Framework can answer a moderate proportion of SRTQs (76%) and have promising performance in retrieving entities from KG. Each column of Table.5.6 has shown that as type or relation expanding the recall increases while the precision decreases. The semantic similarity based type expansion algorithm can improve recall and guarantee a certain level of precision. Since there is no control in relation expansion, though the recall has improved a lot, the precision becomes unacceptable by including too

[7] http://dbpedia.org/snorql/

| settings | wup | lch | res | jcn | lin |
|---|---|---|---|---|---|
| th=0.9 gp1 | (0.33,0.42) | (0.46,0.41) | (0.40,0.42) | (0.40,0.42) | (0.39,0.42) |
| th=0.9 gp2 | (0.003, 0.66) | (0.007,0.66) | (0.004,0.7) | (0.006,0.66) | (0.006,0.66) |
| th=1.0 gp1 | (0.46,0.4) | (0.46,0.41) | (0.41,0.41) | (0.40,0.42) | (0.42,0.40) |
| th=1.0 gp2 | (0.007,0.66) | (0.007,0.66) | (0.005,0.67) | (0.006,0.66) | (0.007,0.66) |

Table 5.6: Average Recall and Precision

many irrelevant entities. Nevertheless, due to significant improvement of the recall, further research will focus on limiting irrelevant entities by automatically filtering those irrelevant relations in order to guarantee the precision. By comparing each row, it has been shown that the semantic similarity measure *lch, jcn* is better in keeping better precision, but with lower improvement of recall. While *wup, res, lin* are promising in improving recall. Fig.4 has shown that decreasing the threshold resulted in tremendous synsets and longer execution time. Further research is also required to keep reducing the irrelevant types and decreasing the execution time.

## 5.3 Summary

In this chapter, we have presented a similarity-based classification framework and semantic entity search framework. In the classification framework, we compare similarity methods in category classification task and show that the similarity-based classification method is more effective comparing to the one-hot representation of feature words. In the search framework, we have shown that the system has promising performance in answering SRTQ and the proposed semantic similarity based type expansion algorithm can improve the entity search recall while keeping certain level of precision. Moreover, it has been shown that the relation expansion in query graph generation has a significant improvement in search recall though precision become unacceptable.

CHAPTER 6

# Conclusions and Future Research

*This thesis has presented a number of solutions that contribute to the area of semantic similarity methods and their applications for KGs. New semantic features contained in KGs are considered in developing those similarity methods and corresponding applications. Especially, we focus on exploiting the similarity methods that can encode both structural knowledge and statistical knowledge in KG, and apply them to KG-based applications including disambiguation, classification and search. In consequence, the achievements of the thesis can help to facilitate the development of KG-based applications and the selection of proper semantic similarity for different applications.*

*In this chapter, those contributions are summarized and final conclusions are presented. Furthermore, having taken account of the achievements of the thesis, this chapter presents possible future research.*

## 6.1 Conclusions

The research presented in this thesis was set out with the goal to propose a solution for the problem of semantic similarity taking account of semantic features in modern large scale KGs. Then, the applications of semantic similarity are studied specially focusing on disambiguation, classification and search. Although many efforts in the KG community are working on the automatic solutions of KG population techniques and higher level KG applications such as QA, the lack of fundamental solutions and tools for developing applications based on constructed KGs motivated us to research on that topic. In the thesis, both knowledge-based and corpus-based similarity methods are investigated and adapted to KGs considering special semantic features that are available in KGs which are concept taxonomy and entity descriptions. We have investigated the structure of concept taxonomy and entity distributions, in order to develop suitable semantic similarity metric for concepts. We have also presented embedding models based on neural network to feed concept taxonomy and entity descriptions for training shared vector space for words and concepts. With word and concept vectors, the similarity between concepts and words are used to develop disambiguation approaches for words and named entities. In conclusion, the discussed similarity methods include concept-concept, concept-word, word-word, word-entity, text-concept and text-entity, while similarity-based applications cover word similarity, WSD, NED, similarity-based classification, and semantic entity search. Taking an overview of the thesis, a number of contributions have been delivered that can be gathered under four main contributions:

**Semantic similarity** The thesis proposed the WPath similarity metric to compute semantic similarity of concepts in KG which combines structural knowledge of concepts in taxonomy (shortest path length) and statistical knowledge of IC based on the entity distribution. The combination not only retains effective shortest path length to represent distance between concepts, but also includes IC to represent the commonality between concepts and improves variability of similarity scores for ranking concepts. Furthermore, since computing corpus-based IC requires concept-annotation corpus and has high computational cost, the thesis proposed graph-based IC to directly compute the IC of concepts based on KGs given that entities have already been annotated with concepts. Together with WPath similarity method and graph-based IC, the semantic similarity of concepts can be computed only based on KGs and are independent of domain corpus. The proposed methods were evaluated in the standard word similarity dataset. The statistical test has shown that the improvement of proposed methods over conventional knowledge-based methods is statistical significant.

**Semantic disambiguation** The thesis proposed unsupervised similarity-based methods for semantic disambiguation of words and named entities. For WSD, Synset2Vec embedding model was proposed to learn concept and word vectors jointly in order to compute concept-word similarity for sense disambiguation. To enrich the training data for embedding, concept expansion was used over the whole concept hierarchy. Thus, more general concepts are embedded closer to more common words, while more specific concepts are embedded with more specific words. The experimental results in fine-grained WSD dataset have concluded that both the concept expansion and Synset2Vec embedding model are effective. Regarding to NED, most common entity features, textual descriptions and semantic categories, have been studied to apply various semantic similarity methods. We proposed word-similarity based NED approach and Category2Vec based NED approach. Correspondingly, word-word similarity and word-category similarity are used to discriminate ambiguous named entities in comparison to conventional IR and LSA approaches. The Category2Vec embedding method is similar to Synset2Vec, which trains word and category embedding based on of entity descriptions and categories. The experimental results in real world dataset of web queries, questions and tweets have demonstrated: (1) the word-word similarity is effective for NED in short texts; (2) Category2Vec embedding is effective and word-category similarity is useful for NED; (3) both knowledge-based and corpus-based semantic similarity methods are effective for NED while combined models can offer complementary views of NED.

**Similarity-based classification** The thesis proposed similarity-based classification for ontological concept classification. Conventional BOW features have one-hot representation, which has vocabulary mismatch problem using lexical matching, and the trained classifier has limited coverage due to the limited feature words. The thesis contributed with a similarity-based classification framework, where feature vector is constructed with similarity scores between input words and feature words. The similarity vector helps to avoid zero values in feature representation when the feature words are not contained in the input text. In this way, the similar words of feature words are considered according to the similarity model based on statistical information of textual corpora, or structural relation provided by ontology such as WordNet. Moreover, since semantic similarity captures hierarchical relation between concepts, similarity-based classification is able to classify those words to corresponding ontological parent concepts. In consequence, the trained similarity-based classifiers can be used in higher level applications such as type recognition or aspect category classification. The thesis proposed and implemented both unsupervised and supervised similarity-based classification framework. The experiments on concept classification have shown the effectiveness of the proposed methods.

**Semantic Entity Search** The thesis also illustrated the higher level application of semantic entity search including some of the previous contributed modules, such as semantic similarity and disambiguation. The entity search framework consists of linking natural language keyword queries to semantic resources contained in KG, and automatic construction of proper SPARQL queries based on pre-defined query patterns. Then entities are retrieved from execution of corresponding SPARQL queries in KG endpoint. Semantic disambiguation is used to identify correct semantic resources and semantic similarity is used to limit concept expansion in constructing SPARQL queries. Note that the semantic expansion is applied based on the hierarchy of concept taxonomy in order to improve the recall while similarity is used as threshold to guarantee the precision. The semantic entity search system was evaluated in real world entity search datasets, and the experimental results have demonstrated the effectiveness of the search system and expansion strategy. In addition, this entity search system is useful to retrieve entity dataset from KGs using friendly keyword interface, which alleviates human efforts in the tedious work of constructing SPARQL queries, given that users are normally not familiar with the resource vocabulary contained in a given KG.

## 6.2 Future Research

The development of this thesis and its contributions to the state of the art in semantic similarity and their applications for KGs, have opened new possibilities for future research. The experiments conducted have delivered proof for usefulness and applicability of similarity methods for particular applications. Additionally, the implemented software frameworks have stimulated development of new ideas for improving the state of the art in KG-based applications. In terms of conclusions for the thesis research, the following lines of future research can be pointed out as follows.

The thesis proposed the semantic similarity methods based on concept taxonomy and concept distribution over entities. The methods were evaluated in common word similarity datasets and the background KGs are WordNet and DBpedia. Since the proposed methods are applicable to other domains and KGs, it would be desirable a further analysis of effectiveness, suitability and applicability in a specific domain and corresponding KG. Moreover, the thesis analyzed and proposed knowledge-based and corpus-based methods separately and optimized their performance in particular applications. Another interesting future research can be the study of optimization techniques to combine both type of similarity methods, such as employing ensemble approaches. Also, the current optimization of specific semantic similarity for applications are mainly empirical, the further research on automatically deciding the semantic similarity parameters is important for application in terms of saving the development efforts. In addition, the current developed semantic similarity framework has implemented the common KG interfaces and semantic similarity metrics, however, there is still missing some other similarity features and metrics. Consequently, the further extension of the similarity framework to more IC metrics, feature-based approaches would be useful to provide a complete comparison of different semantic similarity methods in different ontology.

Besides, the similarity-based unsupervised disambiguation approach could be extended with other disambiguation features. Supervised machine learning techniques are also options to further improve the overall disambiguation performance considering various similarity features. Among all the possibilities of supervised disambiguation methods, we could highlight some of them, such as a combination of entity prominence, context similarity and entity-entity relatedness in a learning to rank framework in order to optimize the NED model from the entity annotation dataset. As the thesis has investigated the different similarity methods for measuring context similarity, such as word-word, word-category, text-text, and text-category, the ensemble methods would be useful to combine all those similarity methods to obtain a better combined model. Moreover, the Synset2Vec and Category2Vec models share the similar idea and implementation, which provide concept and word vectors derived from KG. Since vector representation provides general representation of words and concepts capturing their associations and connectivity, it would be interesting to apply those vectors to other tasks such as clustering, classification and many others. Thus, the further research on application of trained vector representations would be critical in demonstrating the further capability of proposed embedding model in more general applications.

Regarding to similarity-based classification, the thesis has proposed, implemented and evaluated similarity vectors for concept classification. The classification system is showcased in aspect category classification. Both knowledge-based and corpus-based similarity have shown promising performance in the task. Because of the primary illustration of effectiveness in using similarity features, further experiments in various classification domains could be conducted to test the general applicability of the proposed similarity-based classification framework. The basic idea of using similarity as feature for classification is to solve the vocabulary mismatch of BOW representation. The possible application of similarity-based classification includes entity type classification, in which entities are given types having hierarchical relations. In fact, similarity represents semantic correlation between data. Using such correlation in developing supervised classification system captures the closeness of data respect to their meaning. Such similarity pattern is more general and common than boolean features in different dataset and domains. An interesting future research could be the evaluation of similarity-based classification framework in various text classification tasks, whose idea is to test whether similarity patterns can be shared in different classification domains.

In addition, the thesis has presented a demo application of semantic entity search in order to demonstrated the application of various techniques presented in this thesis. The current demo system is based on the DBpedia endpoint and has limited functionality in terms of answering complex natural language queries. The system can be further improved with following researches considering several aspects. Firstly, the mention detection of nouns and proper nouns only works well in formal queries such as questions. The mention detection task is critical of the consequent modules, thus more advanced machine learning approaches for mention detection would be useful for improving the precision of the NEL system. Secondly, while mapping the mentions to semantic resources in KGs, advanced semantic matching techniques are required. The current system uses exact lexical matching which is simple but has low matching performance. Further researches should pay more attention to the semantic matching techniques such as employing deep learning framework for semantic matching. Thirdly, apart from building more completed name dictionary and applying advanced matching techniques, the current similarity-based disambiguation methods should be further improved when the labeled dataset is available. The combination of more disambiguation features and supervised learning methods can help improve the disambiguation performance, which would result in higher performance of the whole entity search system. Finally, the current SPARQL query generation system was designed for searching entities from KGs with simple graph query patterns. Possible future work would be studying the automatic query generation approach using advanced context free grammars for answering more complicated queries.

Concluding the presented lines of future work: the thesis has investigated and proposed solutions for semantic similarity methods and their applications in KGs. As pointed out in the solution outline presented at the beginning of this dissertation, all the contributions are interconnected and dependent on each other with a central component of semantic similarity. Therefore, aside of answering new questions that the thesis rose, future work should investigate further impact of thesis contributions on other applications that rely on the semantic similarity methods, with special interest in its application in various domain KG and applications. Future researches on how the proposed solutions could be combined with other modeling techniques such as graph-based analysis to realize the ambitious idea of information management based on modern KGs.

# Bibliography

Dbpedia: A nucleus for a web of open data. In *The Semantic Web*, volume 4825 of *LNCS*, pages 722–735. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-76297-3.

C. C. Aggarwal. Towards systematic design of distance functions for data mining applications. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 9–18. ACM, 2003.

N. Aggarwal, K. Asooja, H. Ziad, and P. Buitelaar. Who are the american vegans related to brad pitt?: Exploring related entities. In *Proceedings of the 24th International Conference on World Wide Web Companion*, pages 151–154. International World Wide Web Conferences Steering Committee, 2015.

E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, and A. Soroa. A study on similarity and relatedness using distributional and wordnet-based approaches. In *NAACL*, pages 19–27, Stroudsburg, PA, USA, 2009. ISBN 978-1-932432-41-1.

J. C.-C. Alessandro Raganato and R. Navigli. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of EACL 2017, Valencia, Spain*, 2017.

I. Augenstein, A. Gentile, B. Norton, Z. Zhang, and F. Ciravegna. Mapping keywords to linked data resources for automatic query expansion. In *The Semantic Web: ESWC 2013 Satellite Events*, volume 7955 of *LNCS*, pages 101–112. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-41241-7. doi: 10.1007/978-3-642-41242-4_9.

F. Baader. *The description logic handbook: Theory, implementation and applications*. Cambridge university press, 2003.

R. Baeza-Yates, B. Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

A. Ballatore, M. Bertolotto, and D. C. Wilson. Linking geographic vocabularies through wordnet. *Annals of GIS*, 20(2):73–84, 2014.

K. Balog and R. Neumayer. A test collection for entity search in dbpedia. In *36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 737–740, New York, 2013. ACM. ISBN 978-1-4503-2034-4. doi: 10.1145/2484028.2484165.

M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *IJCAI*, volume 7, pages 2670–2676, 2007.

M. Baroni, G. Dinu, and G. Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL (1)*, pages 238–247, 2014.

R. Bekkerman and M. Gavish. High-precision phrase-based document classification on a modern scale. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 231–239. ACM, 2011.

N. J. Belkin and W. B. Croft. Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*, 35(12):29–38, 1992.

Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *journal of machine learning research*, 3(Feb):1137–1155, 2003.

J. Berant, A. Chou, R. Frostig, and P. Liang. Semantic parsing on freebase from question-answer pairs. In *EMNLP*, pages 1533–1544, 2013.

T. Berners-Lee, J. Hendler, O. Lassila, et al. The semantic web. *Scientific american*, 284 (5):28–37, 2001.

C. Bizer, T. Heath, and T. Berners-Lee. Linked data-the story so far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pages 205–227, 2009a.

C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154 – 165, 2009b. ISSN 1570-8268. doi: http://dx.doi.org/10.1016/j.websem.2009.07.002. The Web of Data.

R. Blanco, G. Ottaviano, and E. Meij. Fast and space-efficient entity linking for queries. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 179–188, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3317-7. doi: 10.1145/2684822.2685317. URL `http://doi.acm.org/10.1145/2684822.2685317`.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

A. Bordes, X. Glorot, J. Weston, and Y. Bengio. Joint learning of words and meaning representations for open-text semantic parsing. In *AISTATS*, volume 351, pages 423–424, 2012.

A. Budanitsky and G. Hirst. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and other lexical resources*, volume 2, pages 2–2, 2001.

A. Budanitsky and G. Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

R. C. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL*, volume 6, pages 9–16, 2006.

J. Camacho-Collados, M. T. Pilehvar, and R. Navigli. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, 240:36–64, 2016.

A. E. Cano, D. Preotiuc-Pietro, D. Radovanović, K. Weller, and A.-S. Dadzie. #microposts2016: 6th workshop on making sense of microposts: Big things come in small packages. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW 16 Companion, pages 1041–1042, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-4144-8. doi: 10.1145/2872518.2893528. URL `http://dx.doi.org/10.1145/2872518.2893528`.

D. Carmel, M.-W. Chang, E. Gabrilovich, B.-J. P. Hsu, and K. Wang. Erd'14: entity recognition and disambiguation challenge. In *ACM SIGIR Forum*, volume 48, pages 63–77. ACM, 2014a.

D. Carmel, M.-W. Chang, E. Gabrilovich, B.-J. P. Hsu, and K. Wang. Erd'14: Entity recognition and disambiguation challenge. *SIGIR Forum*, 48(2):63–77, December 2014b. ISSN 0163-5840. doi: 10.1145/2701583.2701591. URL `http://doi.acm.org/10.1145/2701583.2701591`.

D. Ceccarelli, C. Lucchese, S. Orlando, R. Perego, and S. Trani. Learning relatedness measures for entity linking. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 139–148, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2263-8. doi: 10.1145/2505515.2505711. URL `http://doi.acm.org/10.1145/2505515.2505711`.

X. Chen, Z. Liu, and M. Sun. A unified model for word sense representation and disambiguation. In *EMNLP*, pages 1025–1035, 2014.

G. Cheng and Y. Qu. Searching linked objects with falcons: Approach, implementation and evaluation. *Int. J. Semantic Web Inf. Syst.*, 5(3):49–70, 2009.

K. Church and W. Gale. Inverse document frequency (idf): A measure of deviations from poisson. In S. Armstrong, K. Church, P. Isabelle, S. Manzi, E. Tzoukermann, and D. Yarowsky, editors, *Natural Language Processing Using Very Large Corpora*, volume 11 of *Text, Speech and Language Technology*, pages 283–295. Springer Netherlands, 1999. ISBN 978-90-481-5349-7. doi: 10.1007\/978-94-017-2390-9_18.

K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29, March 1990. ISSN 0891-2017.

R. L. Cilibrasi and P. M. B. Vitanyi. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383, March 2007. ISSN 1041-4347. doi: 10.1109/TKDE.2007.48.

R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

M. Cornolti, P. Ferragina, and M. Ciaramita. A framework for benchmarking entity-annotation systems. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 249–260, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2035-1. doi: 10.1145/2488388.2488411. URL `http://doi.acm.org/10.1145/2488388.2488411`.

S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, volume 7, pages 708–716, 2007.

I. Dagan, F. Pereira, and L. Lee. Similarity-based estimation of word cooccurrence probabilities. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 272–278. Association for Computational Linguistics, 1994.

D. Damljanovic, M. Agatonovic, and H. Cunningham. Freya: An interactive way of querying linked data using natural language. In *The Semantic Web: ESWC 2011 Workshops*, volume 7117 of *LNCS*, pages 125–138. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-25952-4. doi: 10.1007/978-3-642-25953-1_11.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41 (6):391, 1990.

M. Dragoni, C. da Costa Pereira, and A. G. Tettamanzi. A conceptual representation of documents and queries for information retrieval systems by using light ontologies. *Expert Systems with Applications*, 39(12):10376 – 10388, 2012. ISSN 0957-4174. doi: http://dx.doi.org/10.1016/j.eswa.2012.01.188.

M. Dredze, P. McNamee, D. Rao, A. Gerber, and T. Finin. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 277–285, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1873781.1873813.

P. Edmonds and S. Cotton. Senseval-2: overview. In *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–5. Association for Computational Linguistics, 2001.

W. Fang, J. Zhang, D. Wang, Z. Chen, and M. Li. Entity disambiguation by knowledge and text jointly embedding. *CoNLL 2016*, page 260, 2016.

R. M. Fano and D. Hawkins. Transmission of information: A statistical theory of communications. *American Journal of Physics*, 29(11):793–794, 1961.

P. Ferragina and U. Scaiella. Fast and accurate annotation of short texts with wikipedia pages. *Software, IEEE*, 29(1):70–75, Jan 2012. ISSN 0740-7459. doi: 10.1109/MS.2011.122.

P. Ferragina and U. Scaiella. Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1625–1628, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0099-5. doi: 10.1145/1871437.1871689. URL http://doi.acm.org/10.1145/1871437.1871689.

J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting* *on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219885. URL http://dx.doi.org/10.3115/1219840.1219885.

L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: The concept revisited. *ACM Trans. Inf. Syst.*, 20(1):116–131, January 2002. ISSN 1046-8188. doi: 10.1145/503104.503110.

W. N. Francis and H. Kucera. Brown corpus manual. *Brown University*, 1979.

M. Francis-Landau, G. Durrett, and D. Klein. Capturing semantic similarity for entity linking with convolutional neural networks. In *Proceedings of NAACL-HLT*, pages 1256–1261, 2016.

A. Freitas, J. G. Oliveira, E. Curry, S. O?Riain, and J. C. P. da Silva. Treo: combining entity-search, spreading activation and semantic relatedness for querying linked data. In *Proc. of 1st Workshop on Question Answering over Linked Data (QALD-1) at the 8th Extended Semantic Web Conference (ESWC 2011)*, 2011.

E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, pages 1606–1611, 2007.

O.-E. Ganea, M. Ganea, A. Lucchi, C. Eickhoff, and T. Hofmann. Probabilistic bag-of-hyperlinks model for entity linking. In *WWW*, 2016.

A. Gattani, D. S. Lamba, N. Garera, M. Tiwari, X. Chai, S. Das, S. Subramaniam, A. Rajaraman, V. Harinarayan, and A. Doan. Entity extraction, linking, classification, and tagging for social media: A wikipedia-based approach. *Proc. VLDB Endow.*, 6(11):1126–1137, August 2013. ISSN 2150-8097. doi: 10.14778/2536222.2536237. URL http://dx.doi.org/10.14778/2536222.2536237.

D. Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive science*, 7 (2):155–170, 1983.

R. Gligorov, W. ten Kate, Z. Aleksovski, and F. van Harmelen. Using google distance to weight approximate ontology matches. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 767–776, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7. doi: 10.1145/1242572.1242676.

J. Guo, G. Xu, X. Cheng, and H. Li. Named entity recognition in query. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information*

*Retrieval*, SIGIR '09, pages 267–274, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-483-6. doi: 10.1145/1571941.1571989. URL http://doi.acm.org/10.1145/1571941.1571989.

S. Guo, M.-W. Chang, and E. Kıcıman. To link or not to link? a study on end-to-end tweet entity linking. *Proceedings of NAACL-HLT*, pages 1020–1030, 2013.

Y. Guo, W. Che, T. Liu, and S. Li. A graph-based method for entity linking. In *IJCNLP*, 2011.

M. B. Habib and M. van Keulen. Need4tweet: A twitterbot for tweets named entity extraction and disambiguation. *ACL-IJCNLP 2015*, page 31, 2015.

B. Hachey, W. Radford, and J. R. Curran. Graph-based named entity linking with wikipedia. In *Web Information System Engineering–WISE 2011*, pages 213–226. Springer, 2011.

B. Hachey, W. Radford, J. Nothman, M. Honnibal, and J. R. Curran. Evaluating entity linking with wikipedia. *Artificial Intelligence*, 194:130 – 150, 2013. ISSN 0004-3702. doi: http://dx.doi.org/10.1016/j.artint.2012.04.005. URL http://www.sciencedirect.com/science/article/pii/S0004370212000446. Artificial Intelligence, Wikipedia and Semi-Structured Resources.

M. Hagen, M. Potthast, B. Stein, and C. Bräutigam. Query segmentation revisited. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 97–106, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0632-4. doi: 10.1145/1963405.1963423. URL http://doi.acm.org/10.1145/1963405.1963423.

X. Han and L. Sun. A generative entity-mention model for linking entities with knowledge base. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 945–954. Association for Computational Linguistics, 2011.

X. Han and J. Zhao. Named entity disambiguation by leveraging wikipedia semantic knowledge. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 215–224, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-512-3. doi: 10.1145/1645953.1645983. URL http://doi.acm.org/10.1145/1645953.1645983.

X. Han, L. Sun, and J. Zhao. Collective entity linking in web text: A graph-based method. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 765–774, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0757-4. doi: 10.1145/2009916.2010019. URL `http://doi.acm.org/10.1145/2009916.2010019`.

Z. S. Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

F. Hasibi, K. Balog, and S. E. Bratsberg. Entity linking in queries: Tasks and evaluation. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, pages 171–180. ACM, 2015.

F. Hasibi, K. Balog, and S. E. Bratsberg. Exploiting entity linking in queries for entity retrieval. In *Proceedings of the 2016 ACM on International Conference on the Theory of Information Retrieval*, pages 209–218. ACM, 2016.

Z. He, S. Liu, M. Li, M. Zhou, L. Zhang, and H. Wang. Learning entity representation for entity disambiguation. In *ACL (2)*, pages 30–34, 2013.

F. Hill, R. Reichart, and A. Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *arXiv preprint arXiv:1408.3456*, 2014.

G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

G. Hirst and D. St-Onge. Lexical chains as representation of context for the detection and correction malapropisms. *WordNet: An Electronic Lexical Database*, 1998.

J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 782–792, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4. URL `http://dl.acm.org/citation.cfm?id=2145432.2145521`.

J. Hoffart, S. Seufert, D. B. Nguyen, M. Theobald, and G. Weikum. Kore: Keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 545–554, New York, NY, USA, 2012a. ACM. ISBN 978-1-4503-1156-4. doi: 10.1145/2396761.2396832.

J. Hoffart, S. Seufert, D. B. Nguyen, M. Theobald, and G. Weikum. Kore: keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 545–554. ACM, 2012b.

J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

I. Horrocks. Ontologies and the semantic web. *Commun. ACM*, 51(12):58–67, December 2008. ISSN 0001-0782. doi: 10.1145/1409360.1409377. URL `http://doi.acm.org/10.1145/1409360.1409377`.

N. Houlsby and M. Ciaramita. *Advances in Information Retrieval: 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, April 13-16, 2014. Proceedings*, chapter A Scalable Gibbs Sampler for Probabilistic Entity Linking, pages 335–346. Springer International Publishing, Cham, 2014. ISBN 978-3-319-06028-6. doi: 10.1007/978-3-319-06028-6_28. URL `http://dx.doi.org/10.1007/978-3-319-06028-6_28`.

E. Hovy, R. Navigli, and S. P. Ponzetto. Collaboratively built semi-structured content and artificial intelligence: The story so far. *Artificial Intelligence*, 194:2 – 27, 2013. ISSN 0004-3702. doi: http://dx.doi.org/10.1016/j.artint.2012.10.002. Artificial Intelligence, Wikipedia and Semi-Structured Resources.

E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 873–882, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

I. Hulpuş, N. Prangnawarat, and C. Hayes. Path-based semantic relatedness on linked data and its use to word and entity disambiguation. In *The Semantic Web-ISWC 2015*, pages 442–457. Springer, 2015.

I. Hulpus, N. Prangnawarat, and C. Hayes. Path-based semantic relatedness on linked data and its use to word and entity disambiguation. In *International Semantic Web Conference*, 2015.

I. Iacobacci, M. T. Pilehvar, and R. Navigli. Sensembed: learning sense embeddings for word and relational similarity. In *Proceedings of ACL*, pages 95–105, 2015.

I. Iacobacci, M. T. Pilehvar, and R. Navigli. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 897–907, 2016.

H. Ji and R. Grishman. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1148–1158. Association for Computational Linguistics, 2011.

J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *Computational Linguistics*, cmp-lg/970(Rocling X):15, 1997.

S. S. Kataria, K. S. Kumar, R. R. Rastogi, P. Sen, and S. H. Sengamedu. Entity disambiguation with hierarchical topic models. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 1037–1045, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0813-7. doi: 10.1145/2020408.2020574. URL `http://doi.acm.org/10.1145/2020408.2020574`.

S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 457–466, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-495-9. doi: 10.1145/1557019.1557073. URL `http://doi.acm.org/10.1145/1557019.1557073`.

T. K. Landauer and S. T. Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211, 1997.

J. J. Lastra-Díaz and A. García-Serrano. A novel family of ic-based similarity measures with a detailed experimental survey on wordnet. *Engineering Applications of Artificial Intelligence*, 46, Part A:140 – 153, 2015a. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2015.09.006. URL `http://www.sciencedirect.com/science/article/pii/S0952197615002067`.

J. J. Lastra-Díaz and A. García-Serrano. A new family of information content models with an experimental survey on wordnet. *Knowledge-Based Systems*, 89:509 – 526, 2015b. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2015.08.019. URL `http://www.sciencedirect.com/science/article/pii/S0950705115003305`.

Q. Le and T. Mikolov. Distributed representations of sentences and documents. In T. Jebara and E. P. Xing, editors, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196. JMLR Workshop and Conference Proceedings, 2014. URL `http://jmlr.org/proceedings/papers/v32/le14.pdf`.

C. Leacock and M. Chodorow. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283, 1998.

M. Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26. ACM, 1986.

O. Levy and Y. Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185, 2014.

O. Levy, Y. Goldberg, and I. Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3: 211–225, 2015.

Y. Li, Z. Bandar, and D. Mclean. An approach for measuring semantic similarity between words using multiple information sources. *Knowledge and Data Engineering, IEEE Transactions on*, 15(4):871–882, 2003. doi: 10.1109/TKDE.2003.1209005.

D. Lin. An information-theoretic definition of similarity. In *Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, 1998. Morgan Kaufmann Publishers Inc. ISBN 1-55860-556-8.

X. Liu, Y. Li, H. Wu, M. Zhou, F. Wei, and Y. Lu. Entity linking for tweets. In *ACL (1)*, pages 1304–1311, 2013.

V. Lopez, M. Fernández, E. Motta, and N. Stieler. Poweraqua: Supporting users in querying and exploring the semantic web. *Semantic Web*, 3(3):249–265, 2012.

M. Mancini, J. Camacho-Collados, I. Iacobacci, and R. Navigli. Embedding words and senses together via joint knowledge-enhanced training. *arXiv preprint arXiv:1612.02703*, 2016.

D. L. McGuinness, F. Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(10):2004, 2004.

O. Medelyan, I. H. Witten, and D. Milne. Topic indexing with wikipedia. In *Proceedings of the AAAI WikiAI workshop*, volume 1, pages 19–24, 2008.

E. Meij, M. Bron, L. Hollink, B. Huurnink, and M. de Rijke. Mapping queries to the linking open data cloud: A case study using {DBpedia}. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(4):418 – 433, 2011. ISSN 1570-8268. doi: http://dx.doi.org/10.1016/j.websem.2011.04.001. URL `http://www.sciencedirect.com/science/article/pii/S1570826811000187`. {JWS} special issue on Semantic Search.

E. Meij, W. Weerkamp, and M. de Rijke. Adding semantics to microblog posts. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 563–572. ACM, 2012.

O. Melamud, J. Goldberger, and I. Dagan. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of CONLL*, 2016.

P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. Dbpedia spotlight: Shedding light on the web of documents. In *7th International Conference on Semantic Systems*, I-Semantics '11, pages 1–8, New York, 2011. ACM. ISBN 978-1-4503-0621-8. doi: 10.1145/2063518.2063519.

R. Mihalcea and A. Csomai. Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, pages 233–242, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-803-9. doi: 10.1145/1321440.1321475. URL http://doi.acm.org/10.1145/1321440.1321475.

R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. Association for Computational Linguistics, 2004.

R. Mihalcea, C. Corley, and C. Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI*, volume 6, pages 775–780, 2006.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013b.

G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11): 39–41, 1995.

G. A. Miller and W. G. Charles. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28, 1991.

G. A. Miller, C. Leacock, R. Tengi, and R. T. Bunker. A semantic concordance. In *Proceedings of the workshop on Human Language Technology*, pages 303–308. Association for Computational Linguistics, 1993.

D. Milne and I. H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 509–518, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-991-3. doi: 10.1145/1458082.1458150. URL http://doi.acm.org/10.1145/1458082.1458150.

D. Milne and I. H. Witten. An open-source toolkit for mining wikipedia. *Artificial Intelligence*, 194:222 – 239, 2013. ISSN 0004-3702. doi: http://dx.doi.org/10.1016/j.

artint.2012.06.007. URL http://www.sciencedirect.com/science/article/pii/S000437021200077X. Artificial Intelligence, Wikipedia and Semi-Structured Resources.

A. Moro and R. Navigli. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. *Proc. of SemEval-2015*, 2015.

A. Moro, A. Raganato, and R. Navigli. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2: 231–244, 2014.

D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

R. Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41 (2):10, 2009.

R. Navigli and S. P. Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193: 217–250, 2012.

R. Navigli, D. Jurgens, and D. Vannella. Semeval-2013 task 12: Multilingual word sense disambiguation. In *Proceedings of SemEval 2013*, volume 2, pages 222–231, 2013.

B. P. Nunes, S. Dietze, M. A. Casanova, R. Kawase, B. Fetahu, and W. Nejdl. Combining a co-occurrence-based and a semantic measure for entity linking. In *The Semantic Web: Semantics and Big Data*, pages 548–562. Springer, 2013.

T. Pedersen, S. Patwardhan, and J. Michelizzi. Wordnet::similarity: Measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, HLT-NAACL–Demonstrations '04, pages 38–41, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1614025.1614037.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, volume 12, pages 1532–1543, 2014.

F. Piccinno and P. Ferragina. From tagme to wat: A new entity annotator. In *Proceedings of the First International Workshop on Entity Recognition &#38; Disambiguation*, ERD '14, pages 55–62, New York, NY, USA, 2014a. ACM. ISBN 978-1-4503-3023-7. doi: 10.1145/2633211.2634350. URL http://doi.acm.org/10.1145/2633211.2634350.

F. Piccinno and P. Ferragina. From tagme to wat: a new entity annotator. In *Proceedings of the first international workshop on Entity recognition & disambiguation*, pages 55–62. ACM, 2014b.

M. T. Pilehvar and R. Navigli. A large-scale pseudoword-based evaluation framework for state-of-the-art word sense disambiguation. *Computational Linguistics*, 2014.

A. Pilz and G. Paaß. From names to entities using thematic context distance. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 857–866, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0717-8. doi: 10.1145/2063576.2063700. URL `http://doi.acm.org/10.1145/2063576.2063700`.

G. Pirró and J. Euzenat. A feature and information theoretic framework for semantic similarity and relatedness. In *International semantic web conference*, pages 615–630. Springer, 2010.

J. Plu, G. Rizzo, and R. Troncy. Enhancing entity linking by combining ner models. In *Semantic Web Evaluation Challenge*, pages 17–32. Springer, 2016.

M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado*, pages 486–495, 2015.

M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S. M. Jiménez-Zafra, and G. Eryiğit. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California, June 2016a. Association for Computational Linguistics.

M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 27–35. Citeseer, 2016b.

J. Pound, I. F. Ilyas, and G. Weddell. Expressive and flexible access to web-extracted data: A keyword-based structured query language. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, pages 423–434, New York, NY, USA, 2010a. ACM. ISBN 978-1-4503-0032-2. doi: 10.1145/1807167.1807214.

J. Pound, P. Mika, and H. Zaragoza. Ad-hoc object retrieval in the web of data. In *Proceedings of the 19th international conference on World wide web*, pages 771–780. ACM, 2010b.

J. Pound, A. K. Hudek, I. F. Ilyas, and G. Weddell. Interpreting keyword queries over web knowledge bases. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 305–314, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1156-4. doi: 10.1145/2396761.2396803. URL `http://doi.acm.org/10.1145/2396761.2396803`.

S. S. Pradhan, E. Loper, D. Dligach, and M. Palmer. Semeval-2007 task 17: English lexical sample, srl and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 87–92. Association for Computational Linguistics, 2007.

K. Q. Pu and X. Yu. Keyword query cleaning. *Proc. VLDB Endow.*, 1(1):909–920, August 2008. ISSN 2150-8097. doi: 10.14778/1453856.1453955. URL `http://dx.doi.org/10.14778/1453856.1453955`.

R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1):17–30, 1989. doi: 10.1109/21.24528.

D. Rao, P. McNamee, and M. Dredze. Entity linking: Finding extracted entities in a knowledge base. In *Multi-source, Multilingual Information Extraction and Summarization*, pages 93–115. Springer, 2013.

L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1375–1384, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-87-9. URL `http://dl.acm.org/citation.cfm?id=2002472.2002642`.

J. Reisinger and R. J. Mooney. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 109–117, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 1-932432-65-5.

P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume*

*1*, IJCAI'95, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1-55860-363-8, 978-1-558-60363-9.

P. Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11(95):95–130, 1999. doi: 10.1613/jair.514.

A. Ritter, S. Clark, O. Etzioni, et al. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. Association for Computational Linguistics, 2011.

M. A. Rodríguez and M. J. Egenhofer. Determining semantic similarity among entity classes from different ontologies. *Knowledge and Data Engineering, IEEE Transactions on*, 15 (2):442–456, 2003. doi: 10.1109/TKDE.2003.1185844.

S. Rothe and H. Schütze. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. *arXiv preprint arXiv:1507.01127*, 2015.

H. Rubenstein and J. B. Goodenough. Contextual correlates of synonymy. *Commun. ACM*, 8(10):627–633, October 1965. ISSN 0001-0782. doi: 10.1145/365628.365657.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

D. Sánchez, M. Batet, D. Isern, and A. Valls. Ontology-based semantic similarity: A new feature-based approach. *Expert Systems with Applications*, 39(9):7718–7728, 2012. doi: 10.1016/j.eswa.2012.01.082.

U. Sawant and S. Chakrabarti. Learning joint query interpretation and response ranking. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1099–1110, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2035-1. doi: 10.1145/2488388.2488484. URL `http://doi.acm.org/10.1145/2488388.2488484`.

M. Schuhmacher and S. P. Ponzetto. Knowledge-based graph document modeling. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 543–552, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2351-2. doi: 10.1145/2556195.2556250.

N. Seco, T. Veale, and J. Hayes. An intrinsic information content metric for semantic similarity in wordnet. In *Proceedings of the 16th European conference on artificial intelligence*, pages 1089–1090. IOS Press, 2004.

S. Shekarpour, S. Auer, A. Ngomo, D. Gerber, S. Hellmann, and C. Stadler. Keyword-driven sparql query generation leveraging background knowledge. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, volume 1, pages 203–210, Aug 2011. doi: 10.1109/WI-IAT.2011.70.

S. Shekarpour, K. Hoffner, J. Lehmann, and S. Auer. Keyword query expansion on linked data using linguistic and semantic features. In *Semantic Computing (ICSC), 2013 IEEE Seventh International Conference on*, pages 191–197, Sept 2013. doi: 10.1109/ICSC.2013.41.

S. Shekarpour, E. Marx, A.-C. N. Ngomo, and S. Auer. Sina: Semantic interpretation of user queries for question answering on interlinked data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2014.

S. Shekarpour, E. Marx, A.-C. N. Ngomo, and S. Auer. Sina: Semantic interpretation of user queries for question answering on interlinked data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 30:39 – 51, 2015. ISSN 1570-8268. doi: http://dx.doi.org/10.1016/j.websem.2014.06.002. Semantic Search.

D. Shen, J.-T. Sun, Q. Yang, and Z. Chen. Building bridges for web query classification. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 131–138, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7. doi: 10.1145/1148170.1148196. URL http://doi.acm.org/10.1145/1148170.1148196.

W. Shen, J. Wang, and J. Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460, Feb 2015. ISSN 1041-4347. doi: 10.1109/TKDE.2014.2327028.

W. Shen, J. Wang, P. Luo, and M. Wang. Linden: Linking named entities with knowledge base via semantic knowledge. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 449–458, New York, NY, USA, 2012a. ACM. ISBN 978-1-4503-1229-5. doi: 10.1145/2187836.2187898. URL http://doi.acm.org/10.1145/2187836.2187898.

W. Shen, J. Wang, P. Luo, and M. Wang. Liege:: link entities in web lists with knowledge base. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1424–1432. ACM, 2012b.

W. Shen, J. Wang, P. Luo, and M. Wang. Linking named entities in tweets with knowledge base via user interest modeling. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 68–76. ACM, 2013.

A. Sil and A. Yates. Re-ranking for joint named-entity recognition and linking. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 2369–2374, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2263-8. doi: 10.1145/2505515.2505601. URL `http://doi.acm.org/10.1145/2505515.2505601`.

P. Singer, T. Niebler, M. Strohmaier, and A. Hotho. Computing semantic relatedness from human navigational paths: A case study on wikipedia. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 9(4):41–70, 2013.

R. S. Sinha and R. Mihalcea. Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In *ICSC*, volume 7, pages 363–369, 2007.

B. Snyder and M. Palmer. The english all-words task. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43, 2004.

J. H. Steiger. Tests for comparing elements of a correlation matrix. *Psychological bulletin*, 87(2):245, 1980.

Y. Sun, L. Lin, D. Tang, N. Yang, Z. Ji, and X. Wang. Modeling mention, context and entity with neural networks for entity disambiguation. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1333–1339, 2015.

K. Taghipour and H. T. Ng. One million sense-tagged instances for word sense disambiguation and induction. In *CoNLL*, pages 338–344, 2015.

G. Tummarello, R. Delbru, and E. Oren. Sindice.com: Weaving the open linked data. In *The Semantic Web*, volume 4825 of *LNCS*, pages 552–565. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-76297-3. doi: 10.1007/978-3-540-76298-0_40.

P. D. Turney. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416, 2006.

P. D. Turney, P. Pantel, et al. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188, 2010.

A. Tversky. Feaures of similarity. *Psychological Review*, 84:327–352, 1977.

C. Unger, L. Bühmann, J. Lehmann, A.-C. Ngonga Ngomo, D. Gerber, and P. Cimiano. Template-based question answering over rdf data. In *21st International Conference on World Wide Web*, pages 639–648, New York, 2012. ACM. ISBN 978-1-4503-1229-5. doi: 10.1145/2187836.2187923.

R. Usbeck, A.-C. Ngonga Ngomo, S. Auer, D. Gerber, and A. Both. Agdistis - graph-based disambiguation of named entities using linked data. In *13th International Semantic Web Conference*. 2014. URL `http://svn.aksw.org/papers/2014/ISWC_AGDISTIS/public.pdf`.

R. Usbeck, M. Röder, A.-C. Ngonga Ngomo, C. Baron, A. Both, M. Brümmer, D. Ceccarelli, M. Cornolti, D. Cherix, B. Eickmann, et al. Gerbil: general entity annotator benchmarking framework. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1133–1143. ACM, 2015.

J. Waitelonis and H. Sack. Named entity linking in# tweets with kea. In *Proceedings of 6th workshop on NEEL Challenge in conjunction with 25th WWW Conference*, 2016.

D. Weiss, C. Alberti, M. Collins, and S. Petrov. Structured training for neural network transition-based parsing. *arXiv preprint arXiv:1506.06158*, 2015.

D. Williams and G. Hinton. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics. doi: 10.3115/981732.981751.

D. Yuan, J. Richardson, R. Doherty, C. Evans, and E. Altendorf. Semi-supervised word sense disambiguation with neural models. In *COLING 2016*, 2016.

L. Yujian and L. Bo. A normalized levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1091–1095, 2007.

P. Zadeh and M. Reformat. Feature-based similarity assessment in ontology using fuzzy set theory. In *2012 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–7, 2012. doi: 10.1109/FUZZ-IEEE.2012.6251266.

Z. Zhong and H. T. Ng. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83. Association for Computational Linguistics, 2010.

Q. Zhou, C. Wang, M. Xiong, H. Wang, and Y. Yu. Spark: Adapting keyword query to semantic search. In *The Semantic Web*, volume 4825 of *LNCS*, pages 694–707. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-76297-3. doi: 10.1007/978-3-540-76298-0_50.

Z. Zhou, Y. Wang, and J. Gu. A new model of information content for semantic similarity in wordnet. In *Future Generation Communication and Networking Symposia, 2008. FGCNS'08. Second International Conference on*, volume 3, pages 85–89. IEEE, 2008.

G. Zhu and C. A. Iglesias. Computing Semantic Similarity of Concepts in Knowledge Graphs. *IEEE Transactions on Knowledge and Data Engineering*, PP(99):1–1, 2016. ISSN 1041-4347. doi: 10.1109/TKDE.2016.2610428.

G. Zhu and C. A. Iglesias. Computing semantic similarity of concepts in knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):72–85, 2017.

W. Y. Zou, R. Socher, D. M. Cer, and C. D. Manning. Bilingual word embeddings for phrase-based machine translation. In *EMNLP*, pages 1393–1398, 2013.

S. Zwicklbauer, C. Seifert, and M. Granitzer. Doser-a knowledge-base-agnostic framework for entity disambiguation using semantic embeddings. In *International Semantic Web Conference*, pages 182–198. Springer, 2016.

# List of Figures

# List of Tables

# Glossary

**LOD** Linked Open Data

**KG** Knowledge Graph

**RDF** Resource Description Framework

**RDFS** Resource Description Framework Schema

**OWL** Ontology Web Language

**URI** Uniform Resource Identifier

**IRI** Internationalized Resource Identifiers

**SKOS** Simple Knowledge Organization System

**DL** Description Logics

**BOW** Bag of Words

**CBOW** Continuous Bag of Words

**LSA** Latent Semantic Analysis

**NLP** Natural Language Processing

**WNFS** WordNet First Sense

**NLP** Natural Language Processing

**IR** Information Retrieval

**QA** Question Answering

**AI** Artificial Intelligence

**IE** Information Extraction

**WSD** Word Sense Disambiguation

**NED** Named Entity Disambiguation

**NEL** Named Entity Linking

**NER** Named Entity Recognition

**NERD** Named Entity Recognition and Disambiguation

**NERC** Named Entity Recognition and Classification

**TF** Term Frequency

**IDF** Inverse Document Frequency

**IC** Information Content

**IDF** Inverse Document Frequency

**LCS** Least Common Subsumer

**PMI** Pointwise Mutual Information

**PPMI** Positive Pointwise Mutual Information

**VSM** Vector Space Model

**LDA** Latent Dirichlet Allocation

**ESA** Explicit Semantic Analysis

**SVD** Singular Value Decomposition

**ABSA** Aspect Based Sentiment Analysis

# APPENDIX A

# Publications

The results of this thesis have produced a number scientific publications in journals and in conference proceedings. The list of those publications is shown below:

## A.1 Journal Articles

- Ganggao Zhu and Carlos A. Iglesias. Sematch: Semantic similarity framework for Knowledge Graphs. Knowledge-Based Systems (2017). ISSN 0950-7051. **Impact Factor**: 3.325 **Q1**.

- Ganggao Zhu and Carlos A. Iglesias. Computing Semantic Similarity of Concepts in Knowledge Graphs. IEEE Transactions on Knowledge and Data Engineering 29.1 (2017): 72-85. ISSN 1041-4347. **Impact Factor**: 2.476 **Q1**.

## A.2 Conference Proceedings

- Oscar Araque, Ganggao Zhu, Manuel García-Amado and Carlos A. Iglesias , Mining the Opinionated Web: Classification and Detection of Aspect Contexts for Aspect

Based Sentiment Analysis, 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), Barcelona, Spain, 2016, pp. 900-907.

- Antonio M. Diaz, Ganggao Zhu, Álvaro Carrera, Carlos A. Iglesias, Oscar Araque. SmartSim: Improving visualization on social simulation for an emergency evacuation scenario. AmILP ECAI 2016

- Ganggao Zhu, and Carlos Angel Iglesias. Sematch: Semantic Entity Search from Knowledge Graph. SumPre-HSWI ESWC. 2015.

# APPENDIX B

# Developed Tool

An open-source software has been developed together with this thesis. It has been used in the validation of the contributions presented in the thesis and has the aim of fostering further research works. Furthermore, researchers can use it to reproduce the experiments and develop new similarity methods and similarity-based applications, while developers could use it to develop demo applications or select proper methods for their products. The software is available in public source code repositories hosted by GitHub web platform. A brief description is provided below.

**Sematch** Semantic Similarity Framework for Knowledge Graphs.

Semantic Similarity is an important metric to quantify how much two objects (e.g. concept, word or entity) are alike to each other respect to their meanings. It has been proven to be beneficial to various applications such as text classification, machine translation, summarization, question answering, case-based reasoning, similarity-based search and recommendation. Sematch is specially designed and implemented for calculating knowledge-based semantic similarity metrics that rely on structural knowledge of taxonomy (e.g. depth, path length, Least Common Subsumer (LCS)), and statistical Information Contents (IC), both corpus-based IC and graph-based IC. In consequence, Sematch differs from corpus based approaches relying on co-occurrence (e.g. Pointwise Mutual Information (PMI)) or distributional similarity (Latent Semantic Analysis (LSA), Word2Vec, GLOVE and etc). Moreover, the increasing availability of linked data has given birth to the notion of Knowledge Graphs (KGs), with popular examples such as DBPedia or YAGO. KGs are novel semantic networks recording millions of concepts (TBox), instances (ABox), and their relationships. As recent efforts have transformed concept taxonomies such as WordNet into concept taxonomies (TBox) in KGs, semantic similarity metrics can be used to compute similarity for concepts and entities in KGs, by exploiting semantic resources such as structural knowledge of the semantic relationships and statistical graph-based IC. Thus, we have developed an integrated framework, called Sematch, to develop, evaluate and apply semantic similarity between concepts, words, and entities for KGs such as WordNet, YAGO, and DBpedia.

Existing similarity tools mainly follow corpus-based approaches (e.g. gensim) or knowledge-based approaches with a specific taxonomy (e.g. NLTK WordNet2). The use of a specific taxonomy is the main barrier to prevent the application of knowledge-based similarity metrics to concepts and entities for KGs. Moreover, existing tools only provide implementation of a number of similarity metrics. These tools do not provide a framework for defining and evaluating seamlessly new similarity metrics. In addition, similarity metrics should be evaluated in real applications with the aim of assessing about its suitability. With such considerations, Sematch aims to offer a holistic framework that provides: (1) general purpose semantic similarity for KGs including concepts, words, and entities; (2) an evaluation framework for word similarity and similarity-based concept classification; (3) similarity-based applications for concept classification and semantic search.

The core module of Sematch is similarity, while taxonomy and SPARQL modules are used to extract taxonomical features and KGs features respectively for defining similarity metrics. To facilitate evaluation of similarity metrics, evaluation module is built with support of extension to new datasets and new metrics, because similarity evaluation shares the same pipeline for comparing computed metrics to human judgments. In order to investigate the performance of similarity metrics in real similarity-based applications, Sematch provides an application module including implementations of similarity-graph based ranking, similarity-based concept classification, and semantic search for concepts and entities. In addition, Natural Language Processing (NLP), Utility, SimGraph, Dataset modules are included to facilitate feature extraction and implementation of other modules.

Sematch is implemented with Python 2.7 and several open source Python libraries. Users can install Sematch using standard pip install from the Python software index PyPI. Sematch uses NLTK to implement its NLP module and WordNet interface is retained and extended with Open Multilingual WordNet6 and YAGO WordNet mappings in order to support multilingual word similarity and YAGO concept similarity. Networkx is used to implement a common concept taxonomy interface and similarity graph-based ranking algorithm using ranking algorithms such as PageRank. The common concept taxonomy is implemented for computing concept similarity in ontology (TBox) such as DBpedia ontology that are not covered by other similarity tools. Furthermore, RDFlib and SPARQLWrapper are used to manage the ontology file (OWL file) and SPARQL queries respectively, hence, structural, textual and statistical features of concepts and entities can be extracted from KG conveniently. Finally, Scipy, Scikit-learn, Numpy are used as scientific computing libraries for implementing evaluation module and similarity-based classification.

Sematch is mainly used to develop, evaluate and apply semantic similarity metrics for KGs. Researchers can use it to develop new similarity metrics while developers could use it to develop demo applications or select proper similarity metric for their products.

Available at: `http://github.com/gsi-upm/sematch`.

**Sematch-Demo** Sematch demo is a showcase of Sematch framework. Concept, word, and entity similarity computations are available in the demo. Moreover, we develop entity search, concept search, text search and semantic search to demonstrate the application of Sematch framework.

Available at: `https://github.com/gsi-upm/sematch-demo`.