SEMANTIC ANALYSIS OF TWITTER CONTENT

by

Yue Feng

B.Eng. University of Electronic Science and Technology of China, 2013

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Applied Science

in the Program of

Electrical and Computer Engineering

Toronto, Ontario, Canada, 2016

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public for the purpose of scholarly research only.

Abstract

Semantic Analysis of Twitter Content

Yue Feng

Master of Applied Science, Electrical and Computer Engineering

Ryerson University, 2016

Semantic analysis is the process of shifting the understanding of text from the levels of phrases, clauses, sentences to the level of semantic meanings. Two of the most important semantic analysis tasks include 1) semantic relatedness measurement and 2) entity linking. The semantic relatedness measurement task aims to quantitatively identify the relationships between two words or concepts based on the similarity or closeness of their semantic meaning whereas the entity linking task focuses on linking plain text to structured knowledge resources, e.g. Wikipedia to provide semantic annotation of texts. A limitation of current semantic analysis approaches is that they are built upon traditional documents which are well structured in formal English, e.g. news; however, with the emergence of social networks, enormous volumes of information can be extracted from the posts on social networks, which are short, grammatically incorrect and can contain special characters or newly invented words, e.g. LOL, BRB. Therefore, traditional semantic analysis approaches may not perform well for analysing social network posts. In this thesis, we build semantic analysis techniques particularly for Twitter content. We build a semantic relatedness model to calculate semantic relatedness between any two words obtained from tweets and by using the proposed semantic relatedness model, we semantically annotate tweets by linking them to Wikipedia entries. We compare our work with state-of-the-art semantic relatedness and entity linking methods that show promising results.

Acknowledgements

Many people have contributed to my work here at Ryerson University. First I thank my supervisor Dr. Ebrahim Bagheri for guiding my research, as well as providing many helpful suggestions throughout my time here.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Semantic analysis has been widely used in the domain of information retrieval since it can effectively contribute to many applications such as search engines, fraud detection, document summarization, and document translation, just to name a few . In this thesis, we focus on two semantic analysis tasks, namely semantic relatedness and entity linking. Researchers have developed various techniques for these two tasks; however, most of the methods are designed for formal and clean texts such as news articles and books, but with the emergence of social networks in the recent years, huge amount of information can be extracted from social networks such as customer reviews, user sentiments, product information and others, but traditional methods may not work well on social network content because the content in social networks has its special characteristics. For example, there is length limitation for posts on Twitter, each tweet has to be less than 140 characters, therefore, people tend to use abbreviations and newly created words to express their intent, which can result in short and noisy content. Hence, there is need to create semantic analysis methods targeted at social network content.

In this work, we select Twitter as our target social network platform in which, over 500 million tweets per day[1] are posted. Hence, Twitter is considered as a source of significant information. In this thesis, we focus on two tasks. First, we focus on semantic relatedness measurement. Semantic relatedness is defined as any form of lexical or functional association between two words that point to the contextual or semantic similarity of those two words regardless of their syntactical difference [11]. For example, based on our experience, we understand that *car* and *wheel* share high relatedness while there is

[1] www.internetlivestats.com/twitter-statistics/

not much relation between *car* and *textbook*. Second, we focus on entity linking. The goal of this task is to process a document, identify the most relevant semantic concepts in that text and connect them to the entries from structured knowledge bases such as DBpedia and Freebase. For instance, given a tweet '*Sears 4Q earnings fall, adj. results top Street*', the goal would be to annotate it with Wikipedia concepts including *Sears*, *Fiscal_year*.

Overall, we propose a novel semantic relatedness method that takes into account the vast information available on Twitter and calculates the relatedness between words obtained from tweets. Moreover, by using the semantic relatedness method we design especially for tweets, we develop a creative entity linking approach targeted at Twitter content. We have performed experimental evaluations in both tasks against state of the art techniques which show promising results. Since we focus on two semantic analysis tasks, we will present the observation, problem statement and contribution for each of them respectively in the following sections.

1.2 Observations

1.2.1 Semantic Relatedness

Researchers have already used many different information and knowledge sources in order to compute the semantic relatedness between words. These sources include WordNet which is an English dictionary created by linguistic experts, Wikipedia which is an online encyclopedia contributed by online users and Google search engine results, just to name a few. For example, Gabrilovich and Markovitch [33] have proposed a popular semantic relatedness method called Explicit Semantic Analysis (ESA) which measures the semantic relatedness between words based on their co-occurrences in the same Wikipedia articles.

Empirical research has already shown that many of the existing semantic relatedness methods can provide reasonable correlation with subjective interpretation of relatedness of two words [133]. Therefore, we can conclude that existing methods are able to effectively model the closeness between two words in traditional information tasks such as searching News articles because these methods rely predominantly on stable information sources such as Wikipedia. However, with the emergence of popular microblogging services such as Twitter which have unique characteristics, e.g. short length and informality, semantic relatedness methods need to be modified to make them suitable for information retrieval tasks in such contexts.

In our empirical work, we proposed a Twitter Space Semantic Relatedness technique

(TSSR). We observed that word interpretation and usage can be different depending on the communication medium, i.e., people tend to use the same words to express different meaning depending on whether they are using them on social networks, e.g. Twitter or formal situation, e.g. Wikipedia. To better illustrate this Table 1.1 lists five pairs of words that have been found to be highly related by the ESA method given that they were frequently seen together on Wikipedia; on the contrary, they were never seen together in our Twitter dataset with 10 million tweets. For instance, according to the ESA metric, the words *precedent* and *law* are highly related, but out of the 2,135 and 94,185 times that these two words were observed in our dataset, they never co-occurred in our Twitter dataset. A similar trend can be observed in Table 1.2 where the words that are not highly related according to ESA are highly correlated on Twitter. For example, the word *movie* and *popcorn* co-occurred very frequently in our Twitter dataset whereas in ESA-based semantic relatedness it is far from being high.

Table 1.1: Sample word pairs with high rank in ESA, low rank on Twitter. [2]

#	Word1	Word2	ESA Rank	TSSR Rank
1	decoration	valor	87	347
2	aluminum	metal	73	275
3	precedent	law	96	279
4	psychology	Freud	34	150
5	physics	proton	86	280

Table 1.2: Sample word pairs with low rank in ESA, high rank on Twitter

#	Word1	Word2	ESA Rank	TSSR Rank
1	cup	coffee	167	7
2	love	sex	195	33
3	drink	eat	98	65
4	life	lesson	244	39
5	movies	popcorn	309	21

Above observations directly lead to the rationale for semantic relatedness method we propose:

[1]Pairs are from the WordSimilarity-353 collection and the ranks are the estimated similarity rank from the 353 pairs, by TSSR and ESA.

[2]Pairs are from the WordSimilarity-353 collection and the ranks are the estimated similarity rank from the 353 pairs, by TSSR and ESA.

1. The communication and writing style on Twitter are very different from traditional communication media. In Twitter, the contexts tend to be short and informal which leads to the fact that many new words do not necessarily have explicit linguistic semantics, e.g. tweetup and attwaction.

2. Tweets include hashtags that further qualify the purpose that the user intended to convey. Many of the hashtags were presented through an informal expression and the semantics are only understood within the context of communities on Twitter that use them, e.g. baddayatheoffice and shareforshare.

3. The meaning of some words can shift when used in different communication media, e.g. informal Twitter conversations; therefore, the meaning of a word used in Twitter might be different from its meaning in formal usage, e.g. Yoyo is a popular playing object (toy), but when used on Twitter, it means "you're on your own".

Based on these issues, performing information retrieval tasks on microblogs, such as searching for relevant tweets, finding similar tweets and identifying trending topics, requires customized semantic relatedness methods that take account into the above considerations.

1.2.2 Entity Linking

In the area of entity linking (semantic annotation), several practical semantic annotation systems have already been introduced, however they are not necessarily suitable to the content of Twitter given the special characteristics of tweets [43, 69].

To address these challenges, researchers have built semantic annotation methods by considering text characteristics, e.g., [30] and [72], among others are specially designed for annotating tweets. The goal of the task is to link a phrase within the tweet to the best Wikipedia entry. The challenging aspect of the semantic annotation process is to correctly entity link ambiguous terms due to the fact that multiple possibilities exist for every ambiguous term. In other words, for the same phrase, multiple senses exist with different meanings (also known as disambiguation options). To better illustrate this, as shown in Table 1.3, the term *Apple* can have 52 different senses on Wikipedia (equivalent to 52 entries on Apple's disambiguation page obtained from Wikipedia[3]). The goal is to identify the most relevant sense in the context of the tweet. To this end, existing techniques consider all of the senses as possible valid disambiguation options.

[3] en.wikipedia.org/wiki/Apple$_{(disambiguation)}$

Table 1.3: Dominant use of Wikipedia senses on Twitter

Term	**Senses defined in Wikipedia**	**Dominant senses in Twitter**
Apple	Malus, Cashew_apple, Custard_apple, Love_apple, Apple_Inc., ... (52 senses)	Apple_(fruit), Apple_Inc.
Java	Java_Sea, Java_Trench, Java_Road, Java_(programming_language), Java_(band), Java_Man, ... (38 senses)	Java_coffee, Java_(programming_language), Java_Sea
Maze	Maze_(novel), Maze_(film), Tina_Maze, Maze_(puzzle), Maze_(band), ... (39 senses)	Maze_(puzzle), Maze_(band)
Balance	Balance_(Van_Halen_album), Balance_(advertisement), ... (35 senses)	Balance_(ability), Balance_(accounting)

Inspired by the early idea from Gale et al. [34] that within a given discourse there is often one main sense for each term and while working with Twitter content, we were able to develop a hypothesis that for a given ambiguous term, and within a given time interval, it is very unlikely that all of the senses of an ambiguous term are equally likely to be observed on Twitter. In other words, we hypothesize that from amongst the available senses of an ambiguous term, there are only a limited set of senses that are actually being used on Twitter. Therefore, for a tweet that consists of a set of ambiguous terms, it would be rational to consider the senses that are frequently observed on Twitter for the purpose of disambiguation as opposed to consider the whole sense set. Further to our example and as we will show later in the thesis, there are primarily two senses for the term *Apple* on Twitter, referring to either the `Apple corporation`[4] or the `Apple fruit`[5] and the other 50 senses are very rarely, if at all, observed on Twitter. Based on this hypothesis, we set out to build a semantic annotation technique for Twitter that would concentrate on resolving ambiguities by considering the main senses observed on Twitter.

To better illustrate the significant difference between the number of senses of an ambiguous term observed from Twitter compared to the total number of senses of each term defined in Wikipedia. In Figure 1.1, we present the 945 ambiguous terms available on the dataset from [72] and compare the number of senses obtained from Wikipedia disambiguation pages and our Twitter corpus. As shown from the figure, the number of senses which are frequently used on Twitter is significantly less than the number of senses formally defined in Wikipedia. Therefore, we aim at performing semantic annotation for tweets by considering only the *dominant senses*, i.e., those senses that are frequently

[4] https://en.wikipedia.org/wiki/Apple_Inc.
[5] https://en.wikipedia.org/wiki/Apple

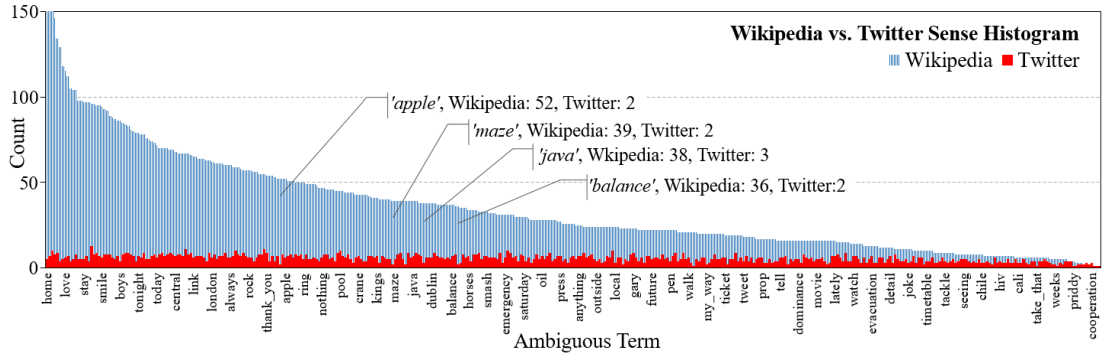

Figure 1.1: Comparison between the number of senses defined in Wikipedia and dominant senses on Twitter

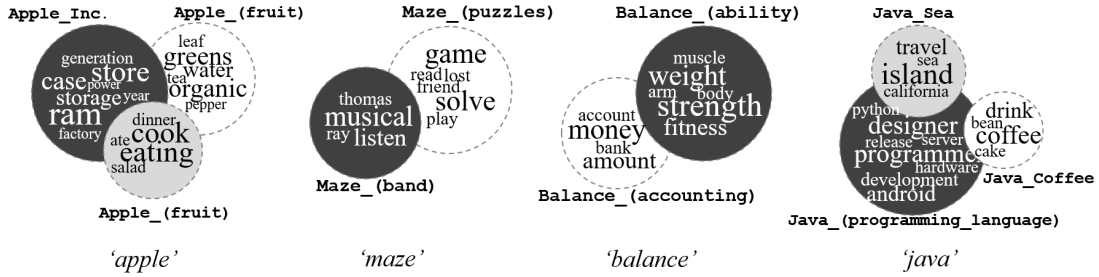

Figure 1.2: Example of sense clusters for ambiguous terms

employed on Twitter. The reason we are interested in identifying dominant senses and only use dominant senses in the annotation process is that by doing so, the annotation process will be much faster and more efficient. We will address two issues in our work:

- Would a semantic annotation technique that only considers the dominant senses extracted from Twitter be able to perform competitively with the state-of-the-art annotation systems that consider the whole sense space in terms of precision and recall?
- Would the consideration of only a limited set of senses significantly reduce the execution time of the tweet annotation process?

In order to be able to identify senses on Twitter, we adopt the *latent relation* hypothesis that states that terms appearing in the same context tend to have related semantics [118]. In the context of Twitter, the hypothesis would mean that the terms that tend to appear together in the same tweets would carry similar or related semantics. As will

be explained further in the thesis, we have identified terms that have related semantics to ambiguous terms based on the latent relation hypothesis. Figure 1.2 shows the most related terms that were found for the ambiguous terms listed in Table 1.3. As shown in the figure, once such semantically related terms are identified, it would be possible to see clusters of highly similar terms to each other that would form the different senses of that term. For instance, Figure 1.2 shows that the dominant senses of the term *Apple* would be the two that were mentioned earlier. Based on this observation, we propose an unsupervised technique that can automatically identify the dominant senses of a term on Twitter. We also present an evaluation of our dominant sense detection method to show that only a small portion of annotation errors was due to the reduced sense set. In other words, eliminating less frequently observed senses on Twitter does not significantly impact the quality of the annotation results.

Based on the *dominant sense* and the *latent relation* hypotheses, the objectives of our work are to *i*) identify the dominant senses of ambiguous terms on Twitter and *ii*) employ the dominant senses to semantically annotate tweets. We evaluate our proposed approach on the publicly available dataset released by Meij et al. [72]. Experimental results show that our method is competitive with other state-of-the-art baselines including supervised and non-supervised approaches in terms of precision and recall despite the fact that we only consider dominant senses of an ambiguous term and ignore the majority of the other senses. Furthermore, we will report that when the tweet that is being processed is temporally aligned with the corpus used for identifying the dominant sense that our approach shows improved performance compared to the state of the art techniques. We also show that our method has a significantly shorter processing time compared to other techniques.

1.2.3 Summary

In this section, we introduce the motivations of our work. First, we intend to build a semantic relatedness measurement method especially for Twitter content because we observe that words usages in traditional content are different than the usages on Twitter due to the nature characteristics of tweets. Second, we hypothesize that people tend to use only a small subset of all the senses of an ambiguous word in daily conversations, therefore we intend to find the dominant senses extracted from tweets and then use these dominant senses to perform tweet annotation.

1.3 Problem Statement

1.3.1 Semantic Relatedness

In the first part of this thesis, we focus on semantic relatedness measurement. We aim to design a method that is suitable in the context of Twitter. To do so, we need to explore the information obtained from Twitter content and represent each target word with the features extracted from tweets, then we can determine the semantic relatedness between two target words by some similarity measures. For example, if we want to calculate the semantic relatedness between word *movie* and word *popcorn* in the context of Twitter, we extract a feature vector for each of the words by mining the information on Twitter which we will introduce in detail in the following sections, then by applying vector similarity methods we can compute the similarity between the two feature vectors where higher similarity indicate higher relatedness between these two words.

To explain the problem more formally, given two words obtained from tweets w_1 and w_2, a feature vector will be built for each of the word such as $\vec{V}(w_1) = \{f_1, f_2, ..., f_n\}$ and $\vec{V}(w_2) = \{f_1^{'}, f_2^{'}, ..., f_n^{'}\}$, then a vector similarity method such as cosine similarity will be applied on above two vectors to produce the final semantic relatedness measure as $SR(w_1, w_2) = cosSim(\vec{V}(w_1), \vec{V}(w_2))$. Since the target words and all the features are extracted from the Twitter content, the proposed method is able to properly identify semantic relatedness of words based on the context of Twitter.

1.3.2 Entity Linking

The main objective of the semantic annotation technique we propose in this thesis is to link a short, noisy and informal tweet to a set of Wikipedia concepts. There are two steps of semantic annotation task: 1) given a tweet, a set of mentions should be detected to be annotated, and 2) the disambiguation process should be conducted to resolve the ambiguity of ambiguous mentions. For example, given a tweet '*Apple has only 2% of India's growing smartphone market. Its quest for more is not going well*', the first step will identify a set of mentions including (*Apple*, *smartphone*) where one of the identified mentions *Apple* is ambiguous, therefore, in the second step, the disambiguation process is needed to find the correct sense of *Apple* in this context which is *Apple corporation*.

Formally speaking, given a tweet T, a collection of mentions is detected as annotations as $M = (m_1, m_2, ..., m_n)$, then the disambiguation process is used to resolve the ambiguity. The final output of the semantic annotation method is a set of Wikipedia entries.

1.4 Contributions

In this thesis, we provide the following contributions:

1. We propose a novel semantic relatedness method which is especially suitable for analyzing the content of Twitter based on our observation that meaning of some words may shift from traditional communication media to social networks.

2. We propose a graph-based method to find the dominant senses of ambiguous terms as used on Twitter by using the semantic relatedness we propose above.

3. We formulate an approach for finding the most suitable Wikipedia sense for each of the identified dominant sense; and

4. We present an annotation technique that entity links a set of ambiguous terms mentioned in a tweet to an unambiguous Wikipedia entry by only taking into account the dominant senses, on-the-fly.

1.5 Structure of the thesis

The structure of the thesis is as follow:

1. Chapter 2 - Literature Review: This chapter covers the details of semantic relatedness methods as well as literature of tweet annotation.

2. Chapter 3 - The Proposed Approaches: This chapter includes two proposed approaches that are Semantic Relatedness Method and Entity Linking Method.

3. Chapter 4 - Empirical Evaluations: In this chapter, two thorough evaluations are performed on the above two proposed approaches respectively and the strengths and weaknesses of these approaches are discussed.

4. Chapter 5 - Conclusion: In this chapter, we summarize the proposed methods and discuss future work.

Chapter 2

Literature Review

In this chapter, we provide an overview of the Semantic Relatedness methods in the literature followed by the state-of-the-art in Entity Linking techniques.

2.1 Semantic Relatedness

Humans can often effortlessly decide about the similarity or relatedness of two words[1]. This can be explained, in part, by the experience that humans have in using and encountering related words in similar contexts. For instance, as human beings, we know *rain* and *umbrella* are highly related, while there is a little, if any, connection between *rain* and *textbook*. While this is trivial for humans, it is often not as simple to translate this judgment process for machines without the careful formulation of background and contextual knowledge surrounding each word and its relationships. Formally speaking, semantic relatedness is defined as a form of semantic or functional associations between two words rather than just lexical relations such as synonymy and hyponymy [11]. The objective of semantic relatedness methods is to closely model such associations.

Semantic relatedness is widely used in many practical applications, particularly in natural language processing (NLP) including semantic information retrieval, keyword extraction, and document summarization, where it is used to quantify the relations between words or between words and documents [62]. Information retrieval techniques have a particular interest in semantic relatedness measures as their incorporation in the retrieval process would allow the identification of meaningfully-related but lexically-dissimilar content [11]. Other more specialized domains such as biomedical informatics

[1]While acknowledging the differences, we use the terms words, concepts, terms and entities, interchangeably in this thesis

and geoinformatics have also benefited from semantic relatedness techniques to identify the relationships between bioentities [96] and geographic concepts [47], respectively.

The development and formalization of semantic relatedness methods is a formidable task that requires solutions for various challenges. In this thesis, we are primarily concerned with two main challenges in this area: (1) challenges related to the underlying knowledge resources that can provide insight into semantic relatedness of words, and (2) challenges related to the formalization of the relatedness measures. In order to understand the scope of these two challenges and to identify the current state of the art, we extensively review work in the area of semantic relatedness, specifically attempting to cover the main models and techniques that have been proposed to address each of the two challenges.

To this end, we propose a taxonomic framework for classifying work in this domain with a specific focus on the above two aspects. The framework is constructed by considering the basic features of semantic relatedness (SR) methods including: (i) the knowledge resources that an SR method adopts; (ii) the computation methods that an SR method is based on; and (iii) the evaluation method that is used to assess the suitability of an SR method including the used datasets and evaluation metrics. The proposed framework dimensions and its sub-dimensions are used as a basis for critically evaluating the strengths and weaknesses of the main work in the domain; consequently, providing a guideline for researchers and practitioners to choose the most appropriate features when constructing or selecting an SR methods according to their needs.

The rest of this section is organized as follows: Section 2.1.1 clearly outlines the criteria used for selecting the methods studied in this thesis and describes each method in detail. Section 2.1.2 presents the proposed framework, and its dimensions and sub-dimensions. Section 2.1.3 compares the selected methods and discusses the strengths and weakness of each method in the context of the proposed framework. Section 2.1.4 concludes the semantic relatedness taxonomy we designed.

2.1.1 Semantic relatedness methods

A typical semantic relatedness method is composed of two main components: its knowledge resource and its computation method. In their evolution, knowledge resources used for semantic relatedness calculation have expanded from the manually constructed lexicon to collaboratively constructed encyclopedia and information available on the World Wide Web. Computation methods have also evolved from the pure statistical analysis, and vector space models to word graph exploration techniques. In this section, we will review fourteen methods from the literature as representatives of the wide range

of methods that have been proposed. Our criteria for selecting these methods are as follows:

- *Selecting methods with a substantial impact on the literature*: Our objective has been to select and review methods that have had a notable impact on the research community. For this purpose, one of the criteria for choosing a study has been its citation count obtained through Google Scholar. We postulate that the higher the citation count for a publication, the better the proposed has been received and recognized by the community.

- *Selecting methods with original proposals*: Our goal has been to include work that was the first to propose an idea with regards to using a knowledge resource or a computation method. The selection included studies that were original work in proposing the idea and not adoptions of earlier ideas. To decide on originality of two similar pieces of work, the work published earlier was chosen as the original one.

For example, we chose ESA [33] since it is the pioneering work in exploring Wikipedias articles and concepts as the underlying knowledge resource. Another example is the work by Sahami and Heilman [103] who was one of the first to propose the use of Web search engine results for developing a similarity kernel function. Table 2.1 shows a summary of the selected methods ordered by their citation counts along with their references, and a brief introduction of each method. As shown later in Section 2.1.3, we have ensured that we have at least one representative method covering each of the framework sub-dimensions.

As shown in Table 2.1, the selected methods utilize a wide range of knowledge resources that have been proposed for semantic relatedness calculation such as Wikipedia, Web search engines, semantic ontologies, and Wiktionary, to name a few. Furthermore, they cover the state of the art methods that are based on word co-occurrences, vector space representation, random walk, the path between words or temporal relation between words. In the remaining part of this section, we give an overview of each selected method.

Resnik [97] hypothesizes that the semantic relatedness between two words is a measure of the amount of information that they share. For this purpose, and in order to identify shared information, the method proposed in [97] identifies the lowest common subsumer of two words within an IS-A hierarchy. The information content value of the subsumer is regarded as an indicator of semantic relatedness.

Table 2.1: Summary of selected methods

Method name	References	Date of publication	Citation count	Brief description
Resnik	[97]	1995	2,675	Considers the information content value of two words based on subsumption relations in a taxonomy
Jiang and Conrath	[55]	1997	2,331	Considers the information content value of two words subsumption relations as well as the information content value of two words in a taxonomy
Lesk	[63]	1986	1,506	Computes the amount of word overlap between the glosses of each word pair
ESA	[33]	2007	1,189	Generates a concept vector to represent each word by exploring related Wikipedia articles
Cilibrasi and Vitnyi	[19]	2007	1,138	Considers the number of pages returned by Google in which the two words co-occur
WikiRelate!	[112]	2006	623	Calculates the length of a path between two nodes in the graph constructed by Wikipedias articles and category tree
Sahami and Heilman	[103]	2006	518	Mines additional information from public Web pages to enhance the representation of a word
Patwardhan and Pedersen	[87]	2006	275	Constructs a second-order gloss vector for each word from Wordnet
Hughes and Ramage	[52]	2007	133	Applies random walk on the graph constructed by exploring the relationship structure of,Wordnet
TSA	[94]	2011	101	Creates a time series concept vector to represent each word by exploring related articles history in Wikipedia
WLM	[83]	2007	99	Constructs vectors for each word by using the links in Wikipedia articles
Zesch et al.	[136]	2008	93	Represents a word using content gathered from the collaboratively-constructed dictionary Wiktionary
Gur	[40]	2005	58	Constructs pseudo glosses for each word by concatenating concepts in close relationship with the word
REWOrD	[91]	2012	4	Makes use of predicates from Semantic Web resources to represent a word

Jiang and Conrath [55] employ the information content value of words as well as the information content value of the two words lowest common subsumer in a lexical taxonomy structure to compute semantic relatedness. The information content value of two words lowest common subsumer describes the amount of information these two words share and the information content value of each word indicates how informative that specific word is. Here, semantic relatedness is defined based on the information content of the lowest common subsumer in the context of the information content of each individual word.

Lesk [63] structures his work on the short pieces of text (glosses) defining each word in WordNet. Specifically, semantic relatedness is computed by counting the number of word overlaps in the glosses of the two words, where higher overlap means higher relatedness between two words.

ESA Gabrilovich and Markovitch [33] have proposed the Explicit Semantic Analysis (ESA) technique which uses Wikipedia as its underlying knowledge resource. The motivation behind ESA is that Wikipedia contains numerous articles and each focuses on a single concept, hence Wikipedia can be viewed as a collection of concepts, each with an article explicitly defined by humans. In their approach, a semantic interpreter is built to map a word into a vector of Wikipedia concepts coupled with weights, where the weights are TF-IDF values of the input word in the underlying articles. In this context, semantic relatedness is measured based on the cosine similarity of the two words vectors.

Cilibrasi and Vitnyi [19] have proposed a method that relies on the information retrieved from a Web search engine. The motivation behind their work is that similar words, when used as search queries, will result in similar Web page results. Therefore, the count of the number of shared Web pages by a Web search engine for three different search queries, namely w1, w2, w1 and w2, is used to formalize the normalized Google distance (NGD). Semantic relatedness is defined as the inverse of NGD.

WikiRelate! [112] takes advantage of Wikipedia articles and category tree to compute semantic relatedness. In their work, the authors apply to Wikipedia the measures that were originally designed for WordNet. Articles are retrieved from Wikipedia by querying word pairs. Wikipedias disambiguation pages obtained for each word are used for disambiguation of the words. The categories related to the retrieved articles are used to compute semantic relatedness by for instance, considering the length of the shortest-path or the length of the path that maximizes information content.

Sahami and Heilman In order to overcome the problem of poor performance that characterizes the traditional document similarity methods when applied on short text snippets, Sahami and Heilman [103] have introduced a new approach for computing the

semantic relatedness. Their method, similar to the work in [19], benefit from Web search results. In particular, they leverage Web search results for enhancing short snippets. Top words ranked based on the TF-IDF measure from the search results are used to build a vector for each input word. The vector is then used to compute the degree of semantic relatedness between two words.

Patwardhan and Pedersen [87] used the co-occurrence information as well as the definitions of words in WordNet to build gloss vectors corresponding to each word. The gloss vector is created in two steps: (1) the first-order vector consisting of co-occurrences between the target word and other words among all the glosses in WordNet is formed; (2) gloss vectors are created by combining the gloss of target words which happen in the first-order vectors. Cosine similarity is applied to the gloss vectors to measure the relatedness between two words.

Hughes and Ramage [52] present an application of Markov chain theory to measure semantic relatedness based on a graph extracted from WordNet. The graph is constructed such that the nodes are words in WordNet and the edges are relational links between words. The authors adopted three types of nodes including *Synset* nodes, *Token-POS* nodes and Token nodes, whereas the relationships types are hypernym/hyponym, instance/instance of, antonym, entails/entailed by, adjective satellite, and causes/caused by. Semantic relatedness is calculated by assuming a particle that starts from a specific word, and then roams through the constructed graph. The particle tends to explore the neighborhood related to the target word, hence resulting in a stationary distribution. Semantic relatedness is the similarity between two stationary distributions obtained for two words.

TSA Radinsky et al. [94] hypothesized that by studying the similarity of word usage patterns over time, a great deal of relatedness information can be discovered to enhance the semantic relatedness results. Thus, they proposed Temporal Semantic Analysis (TSA), which considers temporal information of resources. In their method, each word is represented as a weighted vector of concept time series derived from a historical archive such as NY Times archive. Then semantic relatedness of a pair of words is computed by finding the similarity between two times series representing two words.

WLM In order to reduce the computation expenses of the ESA approach, Milne [83] developed a more efficient method by incorporating links found within corresponding Wikipedia articles. The method assumes that the more links two articles share, the more related they are. Thus, a word is represented as the vector of links. The links are weighted based on a simple but intuitive idea: articles that receive many incoming links can be considered general articles providing less specific information. Semantic similarity

of two words is then the cosine similarity between the weighted vectors representing two words.

Zesch et al. [135] have systematically studied the applicability of Wiktionary as a lexical resource for computing semantic relatedness. They explored the features of Wiktionary including its relation types, languages, size, instance structure, and instance incompleteness in order to propose two semantic relatedness measures namely a path-based approach and a vector-based approach, explained in detail later in the thesis.

Gur The work by Gurevych [40] relies on the structure of GermaNet, a conceptual network of relations between German words. Since GermaNet does not include word glosses for word definitions, Gurevych generated artificial conceptual glosses (pseudo glosses) to describe each word. The pseudo glosses are constructed by concatenating words that are in close relation to the target words through relations such as synonymy, hypernymy, and meronymy, to name a few. Semantic relatedness between two words is then defined as the amount of word overlap between their pseudo glosses.

REWOrD exploits SPARQL queries to access RDF data from DBPedia and evaluates the relatedness of two words based on the informativeness of the path between the two words [91]. The first step in applying REWOrD is to find DBpedia triples whose predicates correspond to each of the words. The informativeness of the predicates is determined based on predicate frequency and inverse triple frequency. The predicates and their informativeness scores are used to build a vector for each word. The cosine similarity between the vectors for the two words is regarded as their degree of relatedness.

2.1.2 Dimensions of the framework

As discussed in the previous section, there are several semantic relatedness approaches and systems in the literature that differ from each other in the way they approach and define relatedness or the resources they use. In this section, we describe various aspects of semantic relatedness techniques based on a classification framework, which consists of three main dimensions and several sub-dimensions. The framework dimensions (Figure 2.1) are as follows:

1. *Knowledge Resources*, including:
   (a) Linguistically constructed resources (Relations, Synsets in WordNet and GermaNet);
   (b) Collaboratively constructed resources (Articles, Article links, Categories, Disambiguation pages in Wikipedia, Information and Relations in English and German Wiktionary);

   (c) Web-based resources (Web Search Engines such as Google, Yahoo,Bing, and the Semantic Web, i.e. the Linked Open Data cloud);

2. *Methods*, including:

   (a) Graph-based methods (Path-based such as Pure Path Length, Graph Length, Common Subsumer, and Random Walk);

   (b) Context-based methods (Co-occurrence-based such as Web Page Hit; Implicit Gloss or Explicit Gloss-based; Vector-based including Gloss Vector, Concept Vector, Links Vector, Predicates Vector, and Feature Vector; Information Content-based such as Concepts Information and Intrinsic Information);

   (c) Temporal methods;

3. *Evaluation Strategies*, including:

   (a) Datasets (English Datasets such as RG-65, MC-30, Fin-353 (Fin1-153,Fin2-200), YP-130. German Dataset like Gur-65, Gur-30, Gur-350, and ZG-222);

   (b) Methods (Correlation with human judgments through Pearson Correlation or Spearman Rank Order Correlation. Application-specific such as Keyphrase Extraction, Semantic Information Retrieval, Word Sense Disambiguation, and Solving word choice problems).

The following subsections describe the framework more deeply covering the three top-level dimensions (Resources, Methods and Evaluation Strategies) and each of their sub-dimensions.

Knowledge Resources

In the context of semantic relatedness techniques, the term knowledge resource refers to the type and source of information that are used for determining the degree of relatedness between two words. We cover three main types of knowledge resources, namely i) linguistically constructed resources such as Wordnet, ii) collaboratively constructed resources such as Wikipedia and iii) Web-based resources including Web search engine results. The taxonomy of the covered knowledge resources is shown in Figure 2.2.

Linguistically Constructed Knowledge Resources: The knowledge resources of this type consist of datasets that have been systematically developed by expert linguists. These knowledge resources are considered the most reliable as they have been curated through a well-reviewed and controlled process. Two of the most widely used resources include WordNet and GermaNet.

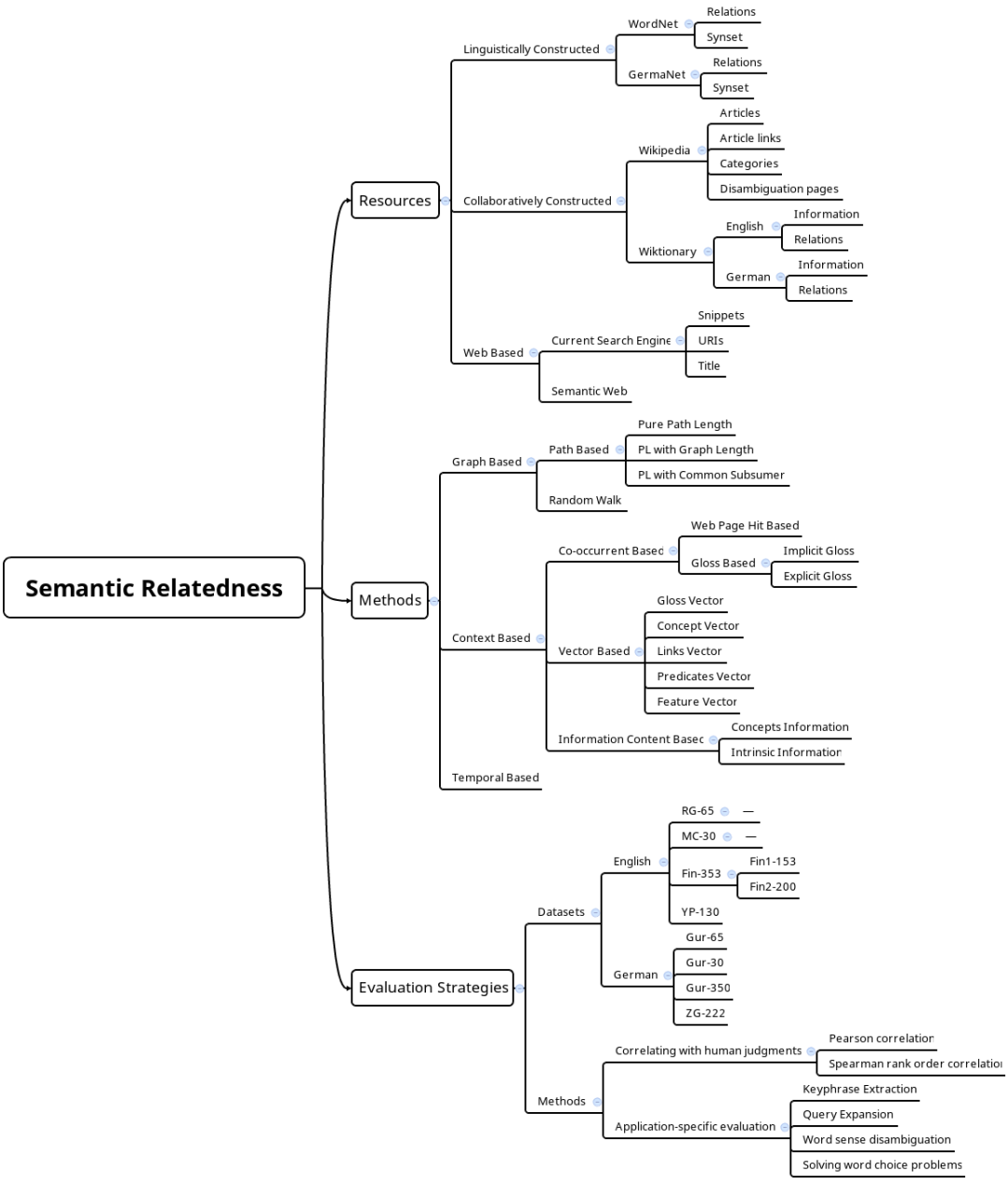

Figure 2.1: The proposed framework dimensions.

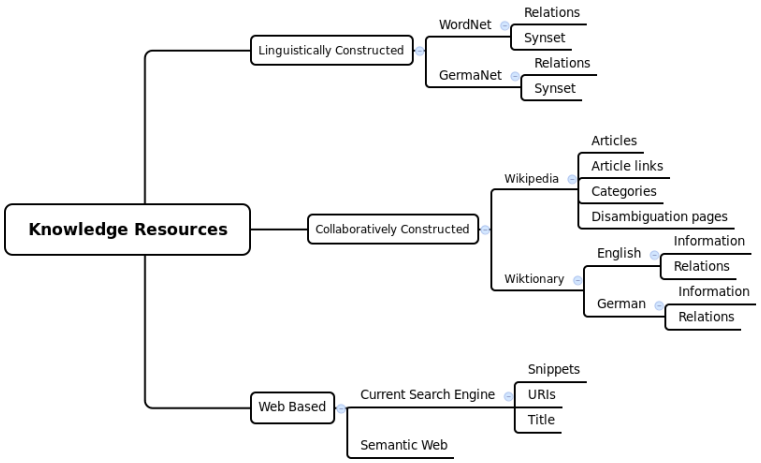

Figure 2.2: Resources taxonomy.

WordNet is a large lexical database for the English language. It consists of information that describes English words and expresses various meanings that a word can have in different contexts. Relations and synsets are two of the main constituents of WordNet where relations express information such as hypernymy, antonymy and hyponymy between two or more words, and synsets represent groups of synonymous words. Additionally, members of each synset are often further described using a short piece of textual descriptor called the gloss. Various researchers have already benefited from Wordnet for computing the degree of semantic relatedness between two words. These works have exploited both Wordnets relations and its glosses.

Relations in Wordnet provide the means to organize words in hierarchical structures. For instance, based on the hyponymy and hypernymy relations, words can be placed in a hierarchy where relations between general and specific terms are explicitly described. This hierarchical structure expressed through Wordnet relations has been the source for various semantic relatedness measures through which the lowest common subsumer of two words has been used as an indication of the relation between them. Resnik [97], Jiang and Conrath [55] and Lin [65] have considered the information content of the subsumer of two words to define the degree of their relatedness. This is based on a simple yet effective observation that subsumers in lower levels of the hierarchy provide more information as they refer to more specific concepts thus revealing greater information content and hence exhibit strong relatedness between two words.

Besides relying on the hyponymy and hypernymy relations, other relationship types

have also been used to create a word graph where the nodes are Wordnet synsets and the corresponding edges are relations. Once the graph is constructed, various graph manipulation techniques have been used to derive relatedness of the nodes in the graph. Rada et al. [93] benefited from this graph representation and represented relatedness as a measure of the shortest path between two nodes. Leacock and Chodorow [60] further improved this calculation by taking into account the depth of the graph along with the path length. In contrast and instead of focusing on path length, Hughes and Ramage [52] applied a random walk process on the graph to extract a statistic distribution, that denotes the probability of reaching other nodes by starting from a given node. Semantic relatedness is then computed by measuring the similarity between two static distributions obtained by starting from each of the two nodes.

While relations in Wordnet allow for identifying structural relatedness between words, glosses allow for the identification of content-based relations between words. A gloss is a short textual definition that describes the meaning of each synset in Wordnet. For example, the gloss of the synset relatedness is a particular manner of connectedness. Various notable work has already been developed that measure semantic relatedness between two words based on the information content overlap of their corresponding glosses. A simple yet effective approach is to count the word overlap between two glosses, and consider the words more related if their word count overlap is higher. While Lesk [63] introduced this method in 1986, some other methods have expanded upon it by introducing the concept of pseudo-glosses. The idea behind pseudo-glosses is that some glosses in Wordnet are too short and hence not effective for calculating relatedness. Therefore, methods are proposed to expand glosses to overcome this problem. Banerjee and Pedersen [4] developed pseudo-glosses for a given word by concatenating the glosses of other related words (e.g. the synset, hypernym, hyponym, holonym, meronym, troponym, and attribute of words in pairs) to its gloss. Mihalcea and Moldovan [76] expanded the glosses by considering the glosses of other words in the WordNet relation hierarchy. Another approach, which deviates from the idea of word overlaps from the glosses, is based on the development of a feature representation for each word where the feature set is created using bags of words within the glosses of the words in WordNet. For instance, Patwardhan and Pedersen [87] represented a word by its second-order gloss vector. In their work, first-order context vectors are created by measuring the co-occurrences between words based on their glosses. Then, the second-order vector for word w is formed by adding the first-order context vectors of words that exist in the gloss of w. For example, the gloss of word fork is cutlery used to serve and eat food, after removing stopwords, the first context vectors of words cutlery, serve, eat and food can be created by counting the

co-occurrences between these words based on their glosses. Then the second-order vector for the word is created by adding the first-order context vectors of these four words.

GermaNet is a German counterpart of WordNet. Many of the approaches applied to WordNet can also be employed for GermaNet. However, the main distinguishing feature of GermaNet is that it does not include glosses; therefore, the original gloss-based methods which calculate relatedness based on glosses are not directly applicable. In order to exploit gloss-based methods, glosses need to be generated from scratch. Gurevych [40] has proposed one such method where pseudo-glosses are generated by concatenating words that are in close relations to the target word in the relationship hierarchy. The generated pseudo-glosses are then used as a representation of the gloss for the words in GermaNet.

Collaboratively Constructed Knowledge Resources: The second class of knowledge resources that are widely exploited in the literature are the information sources that have been collaboratively developed through crowdsourcing on the Web. While these knowledge resources are not necessarily developed by domain expert authorities, they contain reliable information due to extensive peer-review and content moderation. Wikipedia and Wiktionary are amongst the most actively maintained information sources that have received attention from the semantic relatedness community.

Wikipedia: The information collected in Wikipedia is represented through the so-called articles, which are focused on and dedicated to the description of a specific topic. The content of each article is gathered and edited collaboratively and is often strictly moderated by community volunteers. Besides articles, Wikipedia provides hyperlinks between articles, categories, and disambiguation pages. Various researchers have already benefited from the textual content of Wikipedia articles, the hyperlink graph structure as well as categories and disambiguation pages to develop semantic relatedness measures.

One of the widely-used semantic relatedness methods that exploits Wikipedia article content is Explicit Semantic Analysis (ESA) [33]. In this method, each Wikipedia article is assumed to be describing a single word of concepts, which is represented as a weighted mixture of the set of terms that appear in the content of the Wikipedia article. The weights are TF-IDF values of the terms. In ESA, the main idea behind the use of Wikipedia articles is to develop a weighted bag of words representation that can be used for similarity measurement.

Article links, which are inward hyperlinks connecting two Wikipedia articles can be used to establish the relationship between two concepts (words) represented by the two Wikipedia articles. Milne and Witten [83] and Milne [83] have already benefited from article links when proposing the WLM method. They exploit Wikipedia article links by

representing each word as a weighted vector of links computed through the number of links on that words Wikipedia article and the probability of the links occurrence. Different from WLM, WikiWalk [132] exploits Wikipedia article link structure to construct a graph in which Wikipedia articles are the vertices and the edges are the links between the articles. This graph structure, which closely mimics the Wikipedia content structure, is employed for performing a variation of the PageRank algorithm to find word similarities.

The *Wikipedia Category* system is a hierarchical structure where each category can have subcategories through *Hyponymy* or *Meronymy* relations. Each article is coupled with one or more categories. From the category perspective, each category contains one or more articles. Given the meaningful classification that Wikipedia categories provide, WikiRelate [112] defines semantic relatedness between two words based on the mapping between the Wikipedia articles representing the words and their related categories. The basic idea behind this approach is that semantic relatedness of two words is dependent on the relatedness of their categories, therefore, using the mapping, the distance between the categories of two words are taken as a measure of the words semantic relatedness. Other than WikiRelate, WikiWalk [132] also employs Wikipedia category links to augment the graph structure that it builds based on the article links in order to take category similarities into account.

Within Wikipedia, disambiguation pages provide context for words that can have multiple meanings. Disambiguation pages contain links to the most pertinent article per sense of the word along with a brief description. For example, querying *java* returns a Wikipedia disambiguation page which contains links to other pages consisting of the Java Sea, north of the island of Java, Java Trench, a subduction zone trench west of the island of Java, among others. In addition to using Wikipedia categories, WikiRelate also benefits from the disambiguation pages by resolving all redirects in the disambiguation pages and selecting the sense (the redirect link) that results in the highest semantic relatedness between the two words.

Wiktionary: Wiktionary is a multilingual, Web-based, freely available dictionary, thesaurus and phrasebook [135] designed as a lexical companion to Wikipedia. Wiktionary shares many commonalities with Wordnet as they both include words, lexical relations between words and short pieces of text describing the words (glosses). Given the fact that Wiktionary consists of a large number of words, a high dimensional concept vector can be constructed based on its constituent words. For example, Zesch et al. [135] use both English and German versions of Wiktionary to compute semantic relatedness. In their approach, they construct a concept vector $\vec{v}(w) = (v_1, ..., v_n)$ where the value of v_i is the TF-IDF of word w in Wiktionary entry d_i. Once each word is represented

as a concept vector, semantic relatedness between two words is calculated based on the cosine similarity of their concept vectors.

Similar to Wordnet, Wiktionary consists of lexical-semantic relations that are explicitly encoded in the structure of each Wiktionary entry. The English Wiktionary consists of relations such as compounds, abbreviations and acronyms, among others [133]. Some researchers have developed semantic relatedness measures that focus on these relations. As mentioned earlier, the work by Zesch et al. [135] adopts two methods based on Wiktionary content: the first method takes Wiktionary words into account as outlined above and the second method relies on the relations between the words in Wiktionary. In the latter approach, a graph is built whose nodes are the Wiktionary words and the edges are the lexical-semantic relations between these words. Semantic relatedness is then measured by calculating the shortest path between each two nodes. Likewise, Krizhanovsky and Lin [57] have applied a path-based method on a graph constructed based on Russian Wiktionary. In order to address the small vocabulary size of the Russian Wiktionary, the authors have used translations from the Russian Wiktionary to English Wiktionary. On this basis, the shortest path between two words is found and the distance is used to indicate similarity. It is also worth mentioning that Wiktionary has glosses for some of its entries. Therefore, the concept of glosses or more specifically pseudo-glosses can also be exploited for identifying semantic relatedness based on Wiktionary. For example, Meyer and Gurevych [73] explored the glosses in Wiktionary to perform disambiguation based on word overlaps in glosses. They calculated similarity between words with the right sense to create sense-disambiguated word vectors, which resulted in a higher accuracy compared to methods based on WordNet and Wikipedia.

Web based Resources: It has been estimated that there are over 45 billion Web pages on the World Wide Web that have been created with no central coordination[2]. Most of these Web pages carry implicit user-understandable semantics. Many researchers have relied on this implicit semantics to measure semantic relatedness between words. In the Web-based knowledge resource category, two main information sources have been used, namely Web search engines and semantic Web resources.

Web search engines: Given the size of the Web and the role of search engines in content retrieval, there have been extensive researches that have looked at how the results of search engines can be taken as an indication for semantic relatedness. For a given search query, search engines often return useful information such as rich snippets, Web page URIs, user-specified metadata and descriptive page titles. The information content values of the outputs of search engines have been considered as possible indicators of

[2]http://www.worldwidewebsize.com/

relatedness.

Web search engine snippets are short pieces of text for each result returned by the search engine that contain a set of terms that describe the retrieved page. Some authors have benefited from snippets to measure semantic relatedness. For instance, Spanakis et al. [110] have proposed a hybrid Web-based measure for computing semantic relatedness between words by automatically extracting lexico-synactic patterns from snippets based on the idea that similar words should have similar usage patterns. Similarly, Bollegala et al. [9] have developed a semantic relatedness method that relies on search snippets, which considers both word counts and lexical-syntactic patterns when comparing the results of three queries w_1, w_2 and (w_1 and w_2). Sahami and Heilman [103] collect snippets of the top ranked pages for a query and represent each query through the TF-IDF term vector of the collection of the snippets. Semantic relatedness of two words is then computed based on the similarity of their query term vectors. Furthermore, Chen et al. [18] have proposed a double-checking model to analyze snippets returned by a Web search engine, where the double-checking model is formed by a forward process which counts the total occurrences of w_2 in the top N snippets of query w_1 and a backward process which counts the total occurrences of w_1 in the top N snippets of query w_2. Duan and Zeng [26] count the occurrences of each word and also the co-occurrence of the two words within the returned snippets and compute semantic relatedness based on the obtained count frequencies.

There have been other works based on Web search engine results that do not necessarily rely on snippets only, but also consider the content of the retrieved Web pages. The main reason for this is the short length of snippets that could impact the accuracy of the semantic relatedness measures. For example, Sahami and Heilman [103], who initially considered snippets as their knowledge resource, have enhanced snippets by adding the top-k words with the highest TF-IDF values from each of the returned documents to the vector representing each word. Duan and Zeng [26] have also considered the retrieved documents by analyzing the number of documents where the two words occur independently and also co-occur simultaneously. There are several works that operate based on a very similar approach on the retrieved documents, which can be found in [9, 19, 110].

Semantic Web: A more recent knowledge resource is adopted from the Semantic Web community in the form of structured ontologies and the Linked Open Data. The Semantic Web is built primarily on the RDF model which contains triples in the form of <subject, predicate, object>. A triple explicitly defines a relationship between a subject and an object through a meaningful relationship, known as a predicate. As introduced earlier, REWOrD [91] is one of the earlier works that exploit the concept of Linked

Data, especially the DBpedia knowledge base, to compute semantic relatedness. In REWOrD, the correspondence between words and semantic concepts on DBpedia are first found. The retrieved corresponding entities are then used to construct a vector for each word. Vector similarities are then used as the semantic relatedness between two words. Furthermore, Gracia and Mena [37] have calculated semantic relatedness between two concepts within a Semantic Web ontology by finding and comparing the similarity of their ontological contexts. An ontological context for a concept is defined as a collection of highly related concepts within the ontology that can support the unambiguous definition of the concept. For instance, the ontological context can include its hypernyms and synonyms. Karanastasi and Christodoulakis [56] have introduced OntoNL semantic relatedness measure that depends on semantic relations defined by the Web Ontology Language (OWL). In this model, the authors compute semantic relatedness by integrating three aspects: the number of common properties and inverseOf properties that the two concepts share, the path distances of two concepts common subsumer, and the count of the common nouns and synonyms from the concepts descriptions in the ontology. Finally, Zhou et al. [138] have proposed the LODDO method that measures semantic relatedness between words as long as the word has a corresponding concept (entity) on the Linked Open Data. For any given pair of concepts, LODDO would retrieve the description of the concepts from the Linked Open Data cloud and uses text overlap methods to compute the relatedness of two concepts based on their derived descriptions.

Methods

In addition to the knowledge resources used for computing semantic relatedness values, the method that is applied to the adopted knowledge resource plays a significant role in the quality of the relatedness measure. Methods developed for semantic relatedness computation are introduced in this section. The taxonomy of such methods is shown in Figure 2.3. We review three major categories of techniques, namely graph-based, context-based and temporal techniques.

Graph based Methods: The basic idea of graph based methods is to view the information derived from the knowledge resource as a graph whose nodes are terms or concepts, and the edges are some form of relations specific to the selected knowledge resource between pairs of terms or concepts. By adopting a graph-based representation model, many different graph analysis techniques can be applied to the graph to compute semantic relatedness between two words. The two main approaches that have been studied in the literature include path-based methods that consider the path length between two nodes in the graph, and random walk methods that take advantage of the

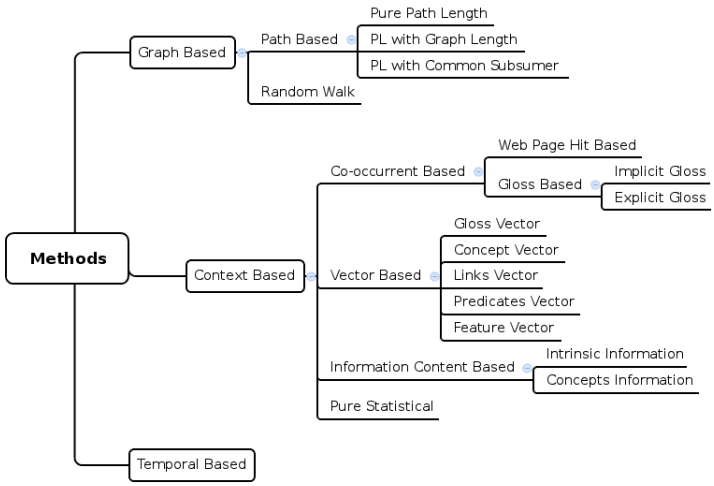

Figure 2.3: Methods taxonomy.

probabilistic likelihood of reaching a destination node from a source node in the graph.

Path based Methods: The path connecting two nodes in the graph-based representation of the knowledge resources can reveal important information about the degree of relatedness between two words. Path based approaches often employ the length of the shortest path between two nodes in the graph to measure their semantic relatedness. It is intuitively assumed that the shorter the path is, the higher the semantic relatedness between the two words would be.

Pure path length methods only consider the length of the shortest path between two nodes which is computed by simply counting the number of edges on the path from one node to the other. For instance, Rada et al. [93] compute semantic relatedness by using the path length l between two nodes where the degree of similarity of two nodes is defined as the length of the longest path in the graph subtracted by l. Jarmasz and Szpakowicz [54] adopt Rada et al's method and apply it to Rogets Thesaurus by counting the number of edges between the two words in Rogets taxonomy. In WikiRelate, Strube and Ponzetto [112] select the shortest path between two words (corresponding to Wikipedia articles) based on the graph constructed from Wikipedia where the nodes are the Wikipedia articles and the edges are the links between the articles. Furthermore, some researchers have used additional corpus statistics in combination with path length to compute semantic relatedness. For example, Jiang and Conrath [55] calculated the sum of all the weights on the shortest path to measure similarity where the weights on the edges are generated from the corpus statistics.

Normalized path length approaches consider additional graph statistics such as the depth of the graph to normalize the length of the shortest path between two nodes. For instance, Leacock and Chodorow [60] normalize the shortest path length by considering the graph depth as $sim_{LG98}(w_1, w_2) = -log\frac{l(w_1,w_2)+1}{2 \cdot depth}$, where $depth$ is the length of the longest path in the graph.

Both pure path length-based and normalized path length-based methods do not consider the information content value of a node. Some researchers have argued that the shared information content value of two words within a graph can be understood through their common subsumer. The consideration of the common subsumer in such approaches ensures that those words which are located higher in the taxonomy (i.e., are more abstract), receive a lower relatedness score compared to those words that are lower in the taxonomy but have comparable path length. For instance, assuming a taxonomic structure, the work by Wu and Palmer [128] is a path length approach, which considers the lowest common subsumer of two words $lcs(w_1, w_2)$ along with the shortest path length between the two words in order to measure semantic relatedness as follows $sim_{WuP94} = \frac{2 \cdot depth(lcs)}{l(w_1,lcs)+l(w_2,lcs)+2 \cdot depth(lcs)}$.

Random Walk Methods: Some researchers have based their semantic relatedness calculation on the likelihood of reaching a node from another node based on a random Markov chain traversal of the graph. In such models, the edges of the graph form a transition matrix between the vertices where each column contains a normalized outgoing probability distribution, and the weight in each cell contains the conditional probability of moving from one node to the other. Based on this initial transition matrix and with repeated conditional transitions, a stationary distribution will be obtained that represents each starting vertex. Semantic relatedness is computed by comparing the similarity between the stationary distributions obtained for two words. For example, Hughes and Ramage [52] construct a graph by extracting information from WordNet where the nodes are *Synsets, TokenPOS* and *Tokens*, the edges are the WordNet relationships between these nodes. The authors define the probability of reaching word w_i at the t^{th} iteration ($w_i^{(t)}$) as the sum of all paths leading to this word on the graph from the previous iteration: $w_i^{(t)} = \sum_{w_j \in V} w_j^{(t-1)} P(w_i|w_j)$. Yeh et al. [132] have applied random walks on Wikipedia link structure for computing semantic relatedness. These authors treated the articles in Wikipedia as vertices and links between articles as edges of the graph. Based on this graph structure, the initial edge weights were determined based on the ESA method [33], after which the Markov chain theory was applied to obtain stationary distributions for each word. Semantic relatedness was then obtained by computing the similarity between any two stationary distributions.

Context based Methods: The latent relation hypothesis postulates that words that are observed in or frequently share similar contexts can be considered to be related [119]. Context-based methods primarily operate based on this hypothesis and attempt to measure semantic relatedness through the degree of similarity of the contexts where words appear in. Different researchers have come up with various forms of word context including Web pages where a word appears in, the Wikipedia articles where a word occurs, and the WordNet glosses where the word is observed, just to name a few. We identify and elaborate on three forms of context-based semantic relatedness methods, namely co-occurrence based, vector-based and information content-based methods.

Co-occurrence based Methods: Two of the most popular contexts that have been commonly used in the literature for this purpose have been the consideration of i) the web pages where the words occur, and ii) WordNet glosses where the words are observed.

In order to exploit the web page content where the words occur, the work proposed in [9, 26, 110] employs a Web search engine to retrieve the specific Web pages where the words occur independently and also simultaneously. The degree of overlap between the retrieved Web pages for each query is used to determine relatedness. Assuming N is the number of documents indexed by the search engine and H(q) be the number of search results for query q, well known set similarity measures such as Jaccard ($\frac{H(w_1 \cap w_2)}{H(w_1)+H(w_2)-H(w_1 \cap w_2)}$), overlap ($\frac{H(w_1 \cap w_2)}{min(H(w_1),H(w_2))}$), Dice ($\frac{2H(w_1 \cap w_2)}{H(w_1),H(w_2)}$), and Point-wise Mutual information ($log_2(\frac{\frac{H(w_1 \cap w_2)}{N}}{\frac{H(w_1)}{N} \times \frac{H(w_2)}{N}})$) are used to measure semantic relatedness of two words w_1 and w_2.

A seminal work in this area is the Google similarity distance proposed by Cilibrasi and Vitànyi [19]. The authors have proposed the normalized Google distance (NGD) to determine the distance between a pair of words where the degree of relatedness is determined based on Googles search results. If two words produce the exact same search result set when used as a query in the Google search engine, their NGD would be zero and if they do not share overlaps, their NGD would be infinite. Gracia and Mena [37] later transformed NGD to compute the relatedness between words regardless of whether Google search is used or not.

As mentioned earlier, context has also been modeled through WordNet glosses where each word's gloss or any gloss where the word is observed are considered to be the context for the word. Many of the existing work such as [40, 63, 135] are based on such context definition and assume that each word has either a WordNet entry with a corresponding gloss or a gloss can be synthetically generated for the word.

When context is modeled as through explicit glosses, the glosses are extracted directly

from WordNet. For example, Lesk [63] built his method by counting the number of word overlaps between two glosses: $—gloss(w_1) \cap gloss(w_2)—$, where $gloss(w_i)$ is the set of words in the gloss of word w_i. Banerjee and Pedersen [4] extended the gloss of each word by taking into account the glosses of related words in order to overcome the problem that some glosses in WordNet are short in length. Moreover, Mihalcea and Moldovan [76] constructed the gloss of a word by combining all the glosses found in its synsets, and then counted the number of word overlaps to determine relatedness.

Considering explicit glosses and their extensions as word context is not always possible, e.g. the case of GermanNet; therefore, in some cases, pseudo glosses are employed as context. For instance, Gurevych [40] constructed pseudo glosses by concatenating words which are in close relation (e.g. Synonymy, Meronymy) with the target word.

Vector based Methods: The idea behind vector based models is to construct a vector representation model for each word that can be used to calculate semantic relatedness through vector similarity measures. Word vectors have been traditionally represented using information extracted from different knowledge resources such as WordNet glosses, Wikipedia links, and Web search result snippets, just to name a few. Based on the type of elements used in the word vector representation, we divide vector based methods into gloss vector, concept vector, link vector, predicate vector and feature vector categories.

Within the gloss vector category, Patwardhan and Pedersen [87] constructed word vectors using WordNet glosses. The authors initially created the first order co-occurrence vectors in WordNet, where the co-occurrences are between the target word and other words in the target word's gloss, and then second order co-occurrences are added to the vector representation, which is inspired by the second order word sense discrimination approach proposed by Schütze [106]. The authors suggest that the use of the Cosine similarity measure on any two such vectors would result in a reliable semantic relatedness value for those two words. Other researchers have also later proposed some variants of the gloss vector representation such as the works by Wan and Angryk [124] and Pedersen [88].

While gloss vector methods focus on the information from WordNet, concept vector methods employ content from Wikipedia to build the vector representation. One of the better known concept vector method, introduced by Gabrilovich and Markovitch [33], is based on the assumption that each Wikipedia article has a topical focus, i.e. the content of each Wikipedia article is focused on a specific topic. Accordingly, a word is represented as a vector whose elements are the TF-IDF values of the words that appear in that article. The limitation of this approach is that it only provides semantic relatedness values between those words that have corresponding Wikipedia articles. Zesch et al.

[135] also created a high dimensional concept vector for each word based on the concept space in Wiktionary.

Unlike gloss and concept vector models, link vector methods represent a word through its links with other words. For this purpose, the link vector model needs to be built on knowledge resources that provide some form of word interlinking, e.g. through hyperlinks. Milne [83] has proposed one of the widely used link vector models where each word is represented by the links that it has to other Wikipedia articles. However, given the fact that not all the links in an article have the same significance, the author defines a weighting scheme for the links. The basis of the weighting scheme is that a page would be considered rather general (less specific) if too many pages link to it. Therefore, Milne defines the weight of a link in a specific Wikipedia article as $log(\frac{N}{|T|})$, where T is the number of articles that link to the target article, and N is the total number of Wikipedia articles. A word is then expressed as a weighted link vector for the links that appear in its corresponding Wikipedia article. Other authors such as Bu et al. [10] and Turdakov and Kuznetsov [116], among others, have also used and promoted the link vector representation.

In the predicate vector representation, the focus is to derive a vector for each word based on the content of RDF documents. For instance, in the REWOrD system, Pirró [91] created a predicate vector for each word, in which the elements of the vector were other words that were connected to the target word through at least one explicit predicate. The author further suggested that the predicate vector could contain other words that are observed along the path of the words that are compared for semantic relatedness. Predicate frequency and inverse triple frequency and path informativeness are metrics that are used to weight each element of the vector.

Finally, feature vector models focus on identifying key discriminative characteristics that can uniquely represent a word. The major difference between feature vector models and the previous three vector representations is that the elements of the feature vector do not rely directly on some form of co-occurrence information but rather they rely on specific metrics to represent a word. For instance, Spanakis [110] proposed to model each word as a feature vector that includes features such as page count metrics, and lexico-syntactic patterns extracted from Google results (e.g. using titles, snippets and URLs). Along the same lines, Bollegala et al. [9] constructed a feature vector based on the lexico-syntactic patterns that they extracted from the results of a Web search, e.g., a word-pair based on the frequency of each pattern. For instance, they determine that words that are related to each other in a given sentence using phrases such as: *also known as, is a, part of, is an example of*, have a high likelihood of being suitable features

for computing semantic relatedness.

Information Theoretic Methods: Information theoretic approaches compute relatedness between words by considering how much common information the two words share. The intuition is that the more similar information the two words convey, the more similar they would be. Information theoretic approaches can be divided into two subcategories depending on how information content sharing is measured.

Intrinsic information theoretic methods rely on a taxonomic knowledge resource for measuring semantic relatedness. To determine the degree of common information shared by two words, intrinsic methods consider features such as position and frequency of the word in the taxonomic structure. For instance, Resnik [97] proposed one of the seminal intrinsic methods where similarity of two words is determined by considering the information content value of two words' subsumer. In his work, Resnik defines information content as the negative log likelihood of the probability of encountering an instance of a given concept. In simple terms, the more general the common subsumer of the words is in the taxonomy hierarchy, the less similar the words would be. Later, Seco et al. [107] base their work on the primary premise of Resniks work by assuming that infrequent words are more informative than frequent ones. Based on this assumption, information content value of a word is determined within the context of WordNet by counting the number of hyponyms that a word has, where the words that have more hyponyms have a lower information content value. Furthermore, the authors assume that words that are leaf nodes in the WordNet hierarchy can be assumed to exert maximal information content.

In the other class of information theoretic approaches, known as information content methods, the information value of the words is considered for computing semantic relatedness. Among the better known works in this class, Jiang and Conrath [55] and Lin [65] have extended the approach developed by Resnik [97] by additionally making use of the information content of a word. In Jiang and Conraths work [55], two measurements were used namely, node-based information content calculation and edge counting. In the node-based approach, the information content of a concept in a taxonomy is defined as the probability that an instance of that concept is encountered in that taxonomy. In the edge counting schema, distance is calculated between two nodes representing instances of the concepts being compared. The shorter the distance is, the more similar the two concepts are. Jiang and Conrath found that the edge counting scheme is highly dependent on the quality of the taxonomy and its structure while the node-based approach is less sensitive to the details of the hierarchy of the taxonomy. Therefore, the authors further proposed an edge-based approach where the distance function between two concepts is

defined as the sum of two concepts information content subtracted from the information content of the concepts' lowest super-ordinate. Furthermore, Lin [65] worked with a tree-structured taxonomy and intuitively assumes that: 1) similarity between two concepts is related to their commonality, where commonality is measured by the number of nodes in the taxonomy that belong to both concepts; 2) similarity between two concepts is also dependent on the differences between them where the difference between two concepts is measured by the number of nodes that exclusively belong to each concept but not the other; and 3) the maximum similarity between two concepts is when they are identical. Lin defined the information content of a concept based on the probability that randomly selected nodes in that taxonomy belong to that concept. Accordingly, semantic relatedness is measured based on the information content of similarities and differences between two concepts.

Temporal Methods: Some researchers have recently focused on the temporal correlation between words to determine their semantic relatedness. While there are not many approaches that consider temporality, the idea behind such approaches is that those words that have similar behavioral patterns over time, e.g. occurrence, can be considered similar. Temporal methods require knowledge resources that incorporate and offer some form of temporality in their information. For instance, Radinsky et al. [94] proposed the Temporal Semantic Analysis (TSA) method where they represented each word as a weighted vector of word time series produced from a historical archive such as the history of Wikipedia articles, which shows the temporal evolution of each article. Based on the time series for each word, the semantic relatedness of two words is measured through time series cross correlation and dynamic time warping. In a different line of work, Milikic et al. [81] have been one of the earlier researchers who have used a non-traditional knowledge resource for temporally modeling semantic relatedness.These authors measured co- occurrence of words on Twitter to calculate semantic relatedness of those words; then, the standard deviation of the semantic relatedness of words within different time periods is employed to estimate the changes of semantic relatedness between words over time. Finally, Zhao et al. [137] hypothesized that temporal factors have a strong impact on the accuracy of similarity measures especially in the context of search queries. Therefore, the authors presented a framework that considers temporal characteristics of historical search click through data to enhance the measurement of similarity between queries. In their work, the similarity between search queries is determined based on the similarity of their historical click through pages over several different time periods.

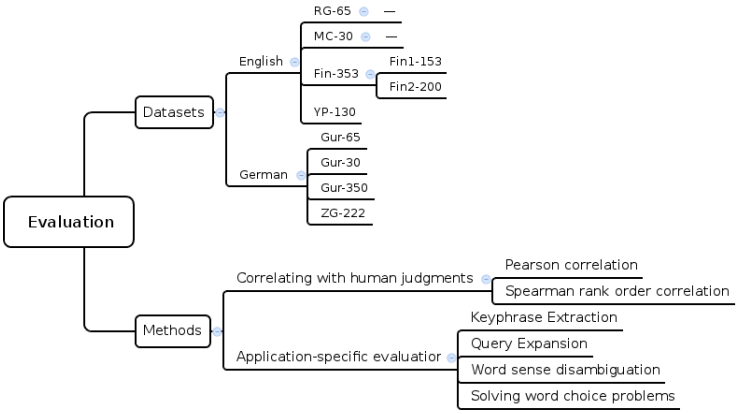

Figure 2.4: Evaluation taxonomy.

Evaluation

In order to evaluate their semantic relatedness work, researchers have used different datasets and methods for comparative analysis. In this section, we focus on classifying the datasets and methods that exist in the literature for evaluating semantic relatedness methods.

The datasets that have been used in the evaluations are mostly curated for the English and German languages. These datasets are often constructed by collecting subjective opinion of humans with regards to the semantic relatedness of words. Table 2.1 provides an overview of some of the common datasets and their brief description. From the perspective of the evaluation methods, these methods can be divided into two main categories, namely determining correlation with human judgments and application-specific evaluations. Table 2.2 provides a summary of the evaluation methods that have been used in the literature. The taxonomy of the datasets and methods used for evaluating semantic relatedness methods is shown in Figure 2.4.

Datasets: The main purpose of developing semantic relatedness datasets is to curate a set of word pairs with known degrees of semantic relatedness so they can be used as a gold standard benchmark for evaluating various semantic relatedness methods. The datasets are most often developed by soliciting human judgments with regards to the semantic relatedness of a set of word pairs. The datasets that have been used and cited in the literature are primarily in the English and German languages.

As shown in Table 2.1 , the most popular English language datasets are the RG-65, MC-30, Fin-353 and YP-130 datasets:

Table 2.2: Summary of datasets used in the literature

DataSet	Language	Date of release	Citation	Number of Pairs	Number of Subjects	Description	SR methods
RG-65	English	1965	[101]	65	51	Includes 65 noun word pairs with score from 0-4	[52], [112], [127], [91], [136], [87]
MC-30	English	1991	[82]	30	38	30 pairs taken from the original RG-65 datasets	[52], [112], [127], [110] , [91], [136], [133], [87], [97], [55], [9]
Fin-353	English	2009	[1]	353	16	Contains 353 English word pairs where 30 word pairs are from MC-30	[52], [112], [33], [127], [110], [91], [136], [133], [83], [94], [26]
YP-130	English	2006	[132]	130	6	Contains 130 verb pairs	[136] [114]
Gur-65	German	2005	[40]	65	24	German translations of the English RG-65	[136] [134] [40]
Gur-350	xGerman	2006	[41]	350	8	Contains 350 word pairs	[136]

1. The Rubenstein Goodenough (RG-65) dataset [101] includes 65 noun pairs, the similarity of each of which is scored on a scale of 0 to 4 where a higher number indicates higher similarity. In order to collect human judgments, 51 subjects participated in the data collection process and the similarity value of each word pair is equal to the average of the values assigned by the subjects. The RG-65 dataset has been used by many researchers as a gold standard to evaluate their semantic relatedness methods, for example, Strube and Ponzetto [112] and Gabrilovich and Markovitch [33] selected RG-65 as a gold standard to analyze their work.

2. The Miller Charles (MC-30) dataset [82] is a subset of 30 pairs taken from the original RG-65 dataset with an additional replicated experiment by another 38 subjects. Given the replicated study and a relatively manageable size of word pairs, the MC-30 dataset has been one of the popular datasets for comparative analytics in many works such as [110, 126].

3. The Finkelstein et al. (Fin-353) dataset [1] contains 353 English word pairs among which 30 word pairs are directly taken from the MC-30 dataset. The dataset is further divided into two subsets where the scores in the first set, Fin-153 (containing 153 word pairs), are obtained from 13 subjects, and in the second set, Fin-200 (containing 200 word pairs), from 16 subjects. Therefore, in some works, the first set has been used for training purposes, and the second set is then used for evaluation. The use of Fin-353 has also been quite popular in the literature including the work by Pirr [91] and Fellbaum [28], among others.

4. The Yang Powers (YP-130) dataset [131] contains 130 verb pairs particularly made for evaluating the ability of a semantic relatedness method to determine the relatedness of verbs. Zesch et al [135] are one of the few works that employed the YP-130 dataset in order to evaluate the ability of their proposed semantic relatedness on verb pairs in addition to more typical noun pairs.

Researchers have also developed datasets in German among which Gur-65, Gur-30, Gur-350 and ZG-222 are the most popular datasets:

1. The Gurevych (Gur-65) dataset [40] is the German translation of the English RG-65 dataset. Gurevych [40] and Zesch et al. [135] have used the Gur-65 dataset to evaluate their methods.

2. The Gurevych (Gur-30) dataset [40] is a subset of the Gur-65 dataset that corresponds to the English MC-30 derived from RG-65.

3. The Gurevych (Gur-350) dataset [41] contains 350 word pairs which includes nouns, verbs and adjectives curated by eight human subjects. Although not heavily used in the literature except in few works, such as [135] , it is a valuable dataset that includes a wide variety of word types that cannot be seen in other datasets.

4. The Zesch Gurvych (ZG-222) dataset [134] consists of word pairs from specific domains. It contains 222 domain specific word pairs that were evaluated by 21 subjects. This dataset also consists of nouns, verbs and adjectives.

Methods: The typical methods for evaluating semantic relatedness techniques can be broadly classified into two classes: 1) computing the degree of correlation with human judgments, and 2) measuring performance in application-specific tasks.

Correlation with human judgments: One of the main techniques for evaluating semantic relatedness methods is to compare their outcomes with a gold standard dataset such as those introduced earlier. Researchers have either compared the absolute predicted relatedness value with the relatedness value of the gold standard, or compared the word pair rankings produced by the relatedness method with the rankings in the gold standard. The latter approach has received more reception as it is less sensitive to the actual relatedness score values and allows for a more pragmatic comparison of the relatedness measures in practice. Such an approach hypothesizes that in order to be considered an accurate semantic relatedness method, the produced rankings from the word pair orderings need to be accurate regardless of the actual numerical value assigned to word pairs. However, in the former evaluation method, the absolute semantic relatedness values are considered to be important with the justification that the rankings in the gold standard datasets do not necessarily accurately represent the desired word pair ordering. This is supported by the fact that in some cases, the gold standard word orderings are sensitive to very small difference between the word pair similarities and therefore, the correct order is questionable.

In order to evaluate the absolute value of the predicted semantic relatedness measure, researchers have predominantly used the Mean Absolute Error (MAE), which measures how closely the predicted value resembles the expected value [5, 92]. Furthermore, for the purpose of measuring rank correlations, Spearmans rank correlation has been used in the literature [33, 52]. Spearman correlation compares if the ranking of the results from the semantic relatedness methods correlates with the ranking provided by human judgments in the gold standard. Pearsons product-moment correlation has also been used by some researchers such as [112].

Application-specific Tasks: As an alternative to the direct evaluation of seman-

Table 2.3: Summary of methods selected by SR methods

Method	SR methods
Pearson correlation	[112]
Spearman correlation	[52], [33], [126], [91] [132], [87], [83], [37], [94], [26]
MAE	[31], [29]
Query suggestion	[103], [122]
Community mining	[9], [16], [77] [70]
Entity disambiguation	[9], [106], [8]
Solving word choice problmes	[135], [54], [117]
Word senses disambiguation	[87], [37], [97] [102]
Ontology matching	[37], [102]
Keyphrase extraction	[84], [133]

tic relatedness methods through a gold standard, application-specific tasks are often used to measure the impact of the proposed semantic relatedness methods on improving the performance of a particular application. The underlying hypothesis of application-specific evaluations is that the more accurate a semantic relatedness measure is, the more it improves the performance of the task at hand. Different authors have used various application-specific tasks for evaluating their work. For instance, Sahami and Heilman [63] evaluated their work in the context of search query suggestion; Bollegala et al. [9] considered the community mining domain to test their semantic relatedness method; Zesch et al. [135], Patwardhan and Pedersen [87], and Gracia and Mena [37] considered entity and word sense disambiguation as their target evaluation application area; Gracia and Mena [37] deployed their method in the context of the ontology matching task. The advantage of application-specific tasks-based evaluation is that not only it shows whether the semantic relatedness measure is able to cause any notable improvement but also shows how well the semantic relatedness measure is suitable for domain specific tasks. For instance, one could show, through experimentation, that although a given semantic relatedness method does not perform well under all conditions, it is effective for a specific task or application area. Table 2.3 shows how different work in the literature have implemented and reported their evaluation strategy and results.

2.1.3 Comparison of the different methods in framework

In this section, we map the selected methods into the proposed framework. To this end, we have created three mapping tables based on the three top level dimensions in the framework: Knowledge Resources (Table 2.4), Methods (Table 2.5) and Evaluation Strategies (Table 2.7). In these tables, the columns show the dimensions and sub-dimensions of our framework and the rows are the methods studied here, and each cell presents the value of the dimension for the selected method.

In order to help researchers or system builders develop their semantic relatedness methods by selecting different features according to their requirements, we summarize the differences, advantages and weaknesses of each dimension in the framework.

Selection of Knowledge Resources

The knowledge resource selected as the underlying foundation for computing semantic relatedness defines primarily how the relationship between the words is viewed. Linguistically constructed knowledge resources accurately model the relations between words and provide reliable definitions of words given they are most often constructed by expert linguists. However, the accurate construction of such knowledge resources are expensive and time consuming and as new words are being added to the language on a constant basis, it is becoming increasingly hard to maintain such resources. Majority of the convered semantic relatedness methods use linguistically constructed knowledge resources due to their accuracy and reliability.

Collaboratively constructed knowledge resources, such as Wikipedia, are created through crowdsourcing. In Wikipedia, articles provide a tremendous amount of information about contexts where certain words appear, the co-occurrence patterns, link structure of content relationships, word sense possibilities and even word and concept categories, which have all been gathered through crowdsourcing. The collaborative nature of such knowledge resources enables the efficient and continuous update of information; therefore, new additions to the language are more likely to be covered. According to a report from Zesch and Gurevych [133] in 2010, the growth of Wikipedia has a positive effect on the coverage without affecting the suitability and accuracy of results. Another unique characteristic of collaboratively constructed knowledge resources is that the involvement of many authors leads to the incorporation of many different distinct styles of writing and word selection, which while may not be ideal for the coherency of the text itself, is an ideal source of information about peoples tendency towards word usage and word relatedness.

Table 2.4: Summary of use of knowledge resource

System	Resource						
	Liguistically Constructured			Collaboratively Constructured		Web Based	
	WordNet	GermaNet	Others	Wikipedia	Wiktionary	Search Engine	Semantic Web
Resnik	*						
Jiang and Conrath	*						
Lesk	*						
ESA				*			
Cilibrasi and Vitanyi						*	
WikiRelate!				*			
Sahami and Heilman						*	
Patwardhan and Pedersen	*						
Hughes and Ramage	*						
TSA				*			
WLM				*			
Zesch et al.					*		
Gur		*					
REWOrD							*

While both linguistically and collaboratively developed knowledge resources provide descriptive information for words, other sources of textual content such as those provided through the Web in general, e.g. Weblogs, news outlets, and social networks, can be used as an informal source of word semantics. Our recent work showed that the semantics of words might shift depending on the context where they are used or where they appear [29]. For instance, there seems to be an observable difference in the most common senses of words when used on Twitter compared to when the words are used on Wikipedia. For this reason, Web content, retrievable through Web search engines, can provide a valuable source of information about word semantics based on their occurrence contexts. However, while this type of resource provides a very high coverage, the accuracy of the information is dependent on the quality of the search engine. The Semantic Web, which provides well-structured and semantically-rich data, can become a useful information resource for determining semantic relatedness. However, the Semantic Web and its associated initiatives such as the Linked Open Data can still be considered an early stage knowledge source that contains considerable noisy data with a relatively small coverage. Table 2.4 summarizes the use of knowledge resource types by the selected methods .

Selection of Computation Method

The selection of the most suitable method for computing semantic relatedness depends on many different factors such as the type of knowledge resource that is adopted, the amount of computing and storage resources available for the computation and the desired accuracy of the approach, just to name a few. For instance, one would only be able to adopt a path-based method if the underlying knowledge resource can be modeled through a graph representation. Furthermore, depending on the type of the path-based method, the explicit type of edges in the graph might also need to have explicit semantics, e.g., in the case of those methods that rely on the common subsumer of two nodes.

Unlike path based methods, random walk based methods do not require e explicit semantics of the relations to be defined in a knowledge resources; they only need the edges to be of the same type and have a quantifiable weight, which could for instance be the co-occurrence number of two words. Therefore, methods that adopt a random walk approach have fewer requirements on the underlying knowledge resource and can be used in conjunction with a wider range of knowledge resources.

Context-based methods can be applied on any knowledge resource that includes minimal description of words; therefore, they are much more flexible and can be used with various types of knowledge resources. For instance, co-occurrence based methods calcu-

late word overlaps in textual information, which can be easily extracted from any source. However, the limitation of such approaches is that information about the various senses of a word is not directly considered and therefore there is a possibility that the usage pattern of ambiguous terms can negatively impact the accuracy of the semantic relatedness scores. One of the pitfalls of the context-based approaches is the role of semantically insignificant words that appear in many different contexts. Such words co-occur with many words and therefore in many cases increase the probability of semantic relatedness between two words that are otherwise not related.

Similar to context-based approaches, vector-based methods do not have specific requirements from the underlying knowledge resource. In such approaches, each word is represented as a vector of features. The most common vector representation is the bag of words model derived from different knowledge resources. When designing vector-based models two important consideration need to be taken into account: i) the bag of words representation for words is extremely sparse and often overlooks word interdependencies. More recent approaches for the vector representation in natural language processing such as Word2Vec [78, 80] and deep semantic embedding [127] can be used to improve this. ii) this model is highly sensitive to the weights of words in the vector [119]; therefore, the decision as to which weighting scheme to be used in the vector would have a high impact on the results. The weighting schemes that require global corpus information would need more computation and update as the corpus evolves. Therefore, while quite straightforward to implement, vector-based models are quite sensitive to features used in the vector representation and the weights applied to the features.

Information-theoretic methods are one of the most restricted models as they are highly coupled with the underlying knowledge resources, which need to have a structured form. The structure of the knowledge resources is used to determine the degree of information that two words share that is used to measure semantic relatedness. Therefore, only knowledge resources such as WordNet can be used in information theoretic methods, thereby, restricting the applicability of such approaches in practice.

Selection of Evaluation Technique

In terms of evaluating the developed semantic relatedness measures, Table 2.7 shows that most authors have adopted a gold standard based approach and compared their results with the gold standard according to the derived ranking of the word pairs using Spearmans rank correlation. As shown in Table 2.6, there are different gold standard datasets that can be used as gold standard. One of the important factors in deciding which gold standard dataset to adopt is the inter-rater agreement of the participants from

Table 2.5: Summary of the discussed methods

System	Methods														
	Graph Based				Context Based										
	Path Based			Random Walk	Co-occurrences Based			Vector Based					Information Theoretic		Temporal
	Pure Path Length	Normalized Path Length	PL with Common Subsumer		Web Page Hit Based	Gloss Based		Gloss Vector	Concept Vector	Link Vector	Predicate Vector	Feature Vector	Intrinsic Information	Information Content	
						Pseudo Gloss	Explicit Gloss								
Resnik													*		
Jiang and Conrath														*	
Lesk							*								
ESA									*						
Cilibrasi and Vitanyi					*										
WikiRelate!	*														
Sahami and Heilman												*			
Patwardhan and Pedersen								*							
Hughes and Ramage				*											
TSA															*
WLM									*						
Zesch et al.								*							
Gur					*										
REWOrD										*					

Table 2.6: Inter-rater agreement values of each datasets

Dataset	Language	InterAA
MC-30	English	0.90
YP-130	English	0.87
Gur-65	English	0.81
RG-65	English	0.80
Fin1-153	English	0.73
Gur-350	German	0.69
Fin2-200	English	0.55
ZG-222	German	0.49
Gur-30	German	-

whom the similarity values were collected. Table 2.6 reports the interrater agreement of the participants for the gold standard datasets where available. As argued by Graham et al. [38], an inter-rater agreement of over 75% would be considered reliable; therefore, gold standard datasets with such agreement or higher can be effectively used in experiments.

One of the reasons that application-specific tasks have not been widely used in the literature is that the accuracy of the semantic relatedness method is not directly observable and is only evaluated indirectly through the performance of the higher level task. Therefore, it is possible that a good performing method is affected by the parameters inside the application framework. In order to use application-specific tasks, controlled experimentation needs to happen where all parameters of the application-specific task are kept constant for the sake of comparison and the semantic relatedness method would be the only variable parameter. The performance of the task would then be measured and directly compared before and after the semantic relatedness method is applied to the task to measure its impact. In summary and according to Table 2.7, for the purpose of evaluation, most authors have chosen to work with RC-65, MC-30, and Fin-353 datasets as their gold standards, in combination with Spearmans rank correlation method.

2.1.4 Summary

In this section, we report on a comprehensive study of semantic relatedness methods, which considers different knowledge resources, methods and evaluations. First, we selected a representative set of semantic relatedness approaches reported in the literature. Then, we created a framework to classify these approaches according to the following three dimensions: knowledge resource(s) used, the method applied for computing relatedness, and the adopted evaluation method. By mapping the selected systems into the framework, we systematically analyzed the advantages and disadvantages of each

Table 2.7: Summary of use of evaluation strategies

System	Evaluation Strategy												
	Datasets						Methods						
	English				German		Correlation with human judgments			Application-specific task			
	RG-65	MC-30	Fin-353	YP-130	Gur-65	Gur-350	Pearson correlation	Spearman rank correlation	Mean Absolute Error(MAE)	Keyphrase Extraction	Query Expansion	Word sense disambiguation	Solving word choice problem
Resnik		*						*					
Jiang and Conrath		*						*					
Lesk													
ESA			*					*					
Cilibrasi and Vitanyi													
WikiRelate!	*	*	*				*						
Sahami and Heilman											*		
Patwardhan and Pedersen	*	*						*				*	
Hughes and Ramage	*	*	*					*					
TSA			*					*					
WLM			*					*					*
Zesch et al.	*	*	*		*	*		*					
Gur					*			*					
REWOrD	*	*	*					*					

identified knowledge resources, relatedness computation method, as well as evaluation methods. Therefore, researchers who would want to further improve or deploy certain semantic relatedness systems or methods can highly benefit from the insight provided by this study.

2.2 Entity Linking

2.2.1 Motivation

With the exponentially increasing amount of data generated on the Web, more useful information can be mined from this resource. However, the data presented on the web is mostly in natural language format which is usually highly ambiguous. On the other hand, many structured knowledge bases have emerged to provide large scale machine-readable content. These knowledge bases provide rich information about the world's entities, their semantic classes, and their mutual relationships [108]. There are many such knowledge bases available including DBpedia [3], YAGO [113], Freebase [7], among others.

Therefore, linking raw data from the web to knowledge bases by finding the corresponding entities becomes a imperative process which is known as entity linking. By applying entity linking technology, various tasks can be achieved such as question answering, query expansion and customer review analysis, just to name a few. Examples of selected applications are introduced in Section 2.2.2.

The task of linking textual mentions to their proper entities from structured knowledge bases has attracted a lot of attention over the past several years [51,68,109,115,130, 139]. This task is primarily composed of two major steps: i) The first step is concerned with the identification of the terms or phrases that have the potential to be linked to some entity in the knowledge base. This involves performing tasks such as term expansion [139], abbreviated form expansion, and domain dictionary lookup [130] to detect misspelled mentions and acronyms. We will introduce some candidate entity generation techniques in Section 2.2.3. ii) The second step deals with assigning a candidate entity to the identified mentions from the first step based on a set of features that measure the relevance of the mention and the candidate entities. There are typically two types of features that have been used in the literature, namely local and global features. Local features include things such as the distance obtained from a cosine similarity measure [68], edit distance similarity [68], the probability of the mention serving as the anchor text for the candidate entity [68] and the temporal relevance of a candidate entity for the given mention [115]. The detail related work will be presented in Section 2.2.4.

2.2.2 Applications

Entity linking is essential for many different tasks, below are several typical applications.

Information Extraction

The common problem of information extraction is named entities extracted are usually ambiguous. A suitable way to solve the problem is linking the named entities to structured knowledge bases. Some works have been proposed in terms of information extraction in this regard. For example, Lin et al. [66] created a technology to link entity mentions extracted from 15 million textual pieces from the web to Wikipedia and they stated that by entity linking, they can find semantically typed textual relations, integrate texts with linked data resources, and perform inference-based rule learning. Another good example is called PATTY [86] which is a taxonomy constructor by exploring relational patterns. To do so, it first employs entity linking techniques to link entities in the extracted relations with YAGO2 knowledge base [49] for the sake of disambiguation.

Information Retrieval

In the field of information retrieval searching based on semantic entities has attracted a lot of attention in the recent years. There are many benefits to adopting entity linking in the search process, for instance, it can disambiguate entity mentions to deal with the semantics of web documents more accurately. For example, the entity mention "java" in a search query could mean many different things, such as the programming language Java, the coffee Java or an island named Java. By bridging the ambiguous entity mentions in the query to unambiguous entities in a knowledge base according to the query context or the user's search history, one could potentially improve the quality of search results.

Content Analysis

General content analysis of text including topic extraction, categorization, event detection, and sentiment analysis can all benefit from the application of entity linking. For example, content-based news recommendation systems [67, 90] link news articles to knowledge bases to explore topics and recommend interesting articles to users. Moreover, besides news articles and web documents, Twitter[3] has become a very popular data source of information due to its rapid growth. Researchers [74] have discovered Twitter users' interests by linking their tweets to a knowledge base.

[3] https://twitter.com/

Question Answering

A question answering system has to leverage its knowledge bases to give the answer to the user's question. Similar to search queries, one of the challenges is to disambiguate entity mentions in the question. For example, to answer the question such as "what is the best book to learn Java?", the first thing the system needs to do is to disambiguate the entity mention "Java". By linking the entity to the "java programming language", it can retrieve the books about it. There are many existing work in this area. For instance, Gattani et al. [35] implemented a user query on kosmix.com through linking entities in the query with a knowledge base. Besides, a famous question answering system called IBM Watson [125] adopted entity linking techniques to predict the answers and obtain promising results.

2.2.3 Candidate Entity Generation

As stated before, the first step of entity linking is identifying the terms or phrases that have the potential to be linked to some entities in the knowledge base. There are three typical types of methods to detect candidate entities which are 1) Name dictionary based methods; 2) Surface form expansion methods and 3) Search engine-based methods. The details of each are introduced in the following subsections.

Name Dictionary Based Methods

Name dictionary based methods are the main techniques to find candidate entities for linking and they are adopted in many works so far such as [39], [35], and [12] among others. A dictionary can be built between various names and their possible mapping entities based on features provided by the knowledge base. If the adopted knowledge base is Wikipedia, then the features that can be explored to create the dictionary include entity pages, redirect pages, disambiguation pages, hyperlinks, and anchor texts on Wikipedia articles. After the dictionary is built, a list of candidate entities can be generated. To be more specific, the name dictionary is a $< key, value >$ mapping, where the *key* is a list of names and the *value* is a list of named entities . Table 2.8 is an example of a part of the name dictionary [108].

Besides leveraging the features from Wikipedia content, there are some studies [14], [17] that explore query click logs and web documents to find entity synonyms, which helps build the dictionary.

After the dicitonary is built, there are several ways to find the candidate entities for an entity name. The common methods adopted by the state-of-the-art include:

Table 2.8: A part of the name dictionary D

k (Name)	k.value (Mapping entity)
Microsoft	*Microsoft*
Microsoft Corporation	*Microsoft*
	Michael Jordan
	Michael I.Jordan
Michael Jordan	*Michael Jordan (footballer)*
	Michael Jordan (mycologist)
	...
Hewlett-Packard Company	*Hewlett-Packard*
HP	*Hewlett-Packard*
Bill Hewlett	*William Reddington Hewlett*

1. The entity name is fully contained in or contains the entity mention.
2. The entity name exactly matches the first letters of all words in the entity mention.
3. Then entity name shares some characters with the entity mention.
4. The entity name has a strong degree of similarity with the entity mention.

Surface Form Expansion Methods

Due to the reason that some entity mentions are acronyms or part of their full names, surface form expansion techniques are applied to identify other possible expanded variations (such as the full name) from the corresponding documents where the entity mention appears. Then the expanded forms of entity mentions can be used in other methods to generate the candidate entity such as the name dictionary based methods introduced before.

Search Engine Based Methods

Some entity linking systems exploit the whole web documents returned by search engines to identify candidate entities. Such work can be found in [46, 61]. Specifically, Han and Zhao [46] queried the Google search engine by the entity mention and its short context and included webpages within Wikipedia as candidate entities returned by the search engine. Lemann et al. [61] also obatained the Wikipedia results returned by the Google search engine and filtered results whose Wikipedia titiles are not significantly Dice or acronym-based similar to the query [108].

Wikipedia can also be treated as a search engine to find relevant entities. Zhang et al. [136] generated a list of candidate entities by querying Wikipedia using the string of the entity mention.

2.2.4 Candidate Entity Ranking

After the previous steps, a list of candidate entities are retrieved, which need to be ranked in decreasing order of relevance to the context. In order to rank the candidate entities, we need to first extract some features. The features can be divided into two typical categories that are 1) local feature and 2) global feature.

Global features take a more comprehensive view towards candidate entity ranking where the relation between the candidate entities for the different mentions of the text are taken into consideration. For instance, Liu et al. [68] introduced a collective inference model to link mentions in a tweet to entities from a knowledge base. The authors integrated two sets of global features to train their collective inference model, namely the entity-to-entity and the mention-to-mention similarity features. Through the use of these two sets of features, the authors tried to link similar mentions to similar entities while preserving the high total similarity between matched mention-to-entity pairs. They consider the inter-entity link structure amongst the pairs of entities on Wikipedia as a measure for entity-to-entity similarity. The mention-to-mention set of features consist of the textual similarity between pairs of tweets and whether they are from the same author. In order to combine the above-mentioned three sets of features, the authors employ a greedy hill-climbing approach in the training process to learn the best weighting coefficients for each of the features. In [139], Zou et al. employed belief propagation methods based on topic distribution instead of common links to calculate the global features. The reason is that common links between entities could imply content similarity and subsequently, similar topic distribution.

Similarly, a recent study by Li et al. [64] intentionally removed the cross-links between the entities in the knowledge base from consideration. They proposed a generative model instead, relying solely on textual content, to associate a mention to an entity in a linkless knowledge base. TagMe[4] [30] is one of the better known semantic annotation tools, which has been specifically built for short text, and has shown to perform reasonably well on different datasets and for various benchmark metrics [21]. TagMe uses Wikipedia anchor texts and pages to cross reference short text fragments with Wikipedia articles. Similar to the idea of global features, TagMe benefits from collective agreement between

[4]https://tagme.d4science.org/tagme/

the entity associated with a mention and all of the other entities detected in the text. Different from TagMe, the work by Meij et al. [72] performed entity linking by learning the importance of three types of features, i.e., n-gram features, concept features, and tweet features, in the linking task. In order to achieve this objective, the authors use various machine learning techniques that are trained on a training set using ten-fold cross validation. The authors show that random forests or gradient boosted regression trees can improve the precision of the linking task.

Most, if not all, of the above work do not consider the fact that the choice for the most appropriate entity for a given mention could be influenced by time. In other words, these approaches build probability distributions based on the entity and text co-occurrence within the source corpus, e.g. Wikipedia, and use these distributions to calculate the local and global features. Therefore, these models will not be able to use dynamic information about the temporal co-evolution of mentions and entities. Tran et al. [115] is one of the only few that considers the notion of temporality. The authors incorporate temporal information from the Wikipedia edit history and page view logs to link hashtags to entities. For instance, while '#sochi' refers to a city in Russia, the hashtag was used to report the *2014 Winter Olympics* during February 2014. In our work, we also consider the notion of temporality as we determine the dominant senses of terms on Twitter within certain time periods.

For the purpose of determining the correct entity, some approaches adopt a graph-based representation for the inter linking of local and global features. Shen et al. [109] also turn the tweet entity linking problem into a user-oriented graph-based interest propagation problem. They assume each user has a constant underlying topic interest distribution over various named entities and propose KAURI to collectively link mentions in all tweets posted by the user to the users topics of interest. In a similar vein, Huang et al. [51] propose a graph regularization model to collectively identify and at the same time disambiguate mentions within a tweet. As we will mention, this work is the only work in the literature that employs a semi-supervised method for tweet annotation.

From a training perspective, the annotation models can be classified as supervised and unsupervised. Unsupervised models build probability distributions based on the characteristics of the source corpus. TagMe [30] and DBpedia Spotlight [24] are some examples of unsupervised methods. Such approaches would, therefore, perform in the same way regardless of the input tweets that need to be annotated. Supervised models; however, are trained and fine-tuned based on an initial set of labeled tweets; there- fore, would perform more suitably for the set that they are trained on. The work by Meij et al. [72], Liu et al. [68] and Wikify! [75] are examples of supervised techniques. Huang et al.s

work is the only work that has considered the semi-supervised approach for annotation and has reported competitive performance compared to supervised approaches with only 50% labeled data.

2.2.5 Evalution

In this section, we introduce methods and datasets that have been used in the literature for evaluating entity linking systems.

Evaluation Methods

The entity linking problem is often treated as a ranking problem, therefore, the evaluation measures, such as *precision*, *recall*, $F_1 - measure$, and *accuracy* are frequently used to evaluate an entity linking system. The *precision* of an entity linking system is computed as the fraction of correctly linked entity mentions that are generated by the system:

$$precision = \frac{|correctly\ linked\ entity\ mentions|}{|lined\ mentions\ generated\ by\ system|} \tag{2.1}$$

Precision considers all the entity mentions detected by the system and determines how correct the system is. Besides precision, *recall* is usually used along with the *precision* which is the fraction of correctly linked entity mentions that should be linked:

$$recall = \frac{|correctly\ linked\ entity\ mentions|}{|entity\ mentions\ that\ should\ be\ linked|} \tag{2.2}$$

Recall considers all the entity mentions that should be linked. In order to combine *precision* and *recall*, there is a measure called $F_1 - measure$ which provides a single score to evaluate a system defined as follows:

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{2.3}$$

Besides, the standard performance metrics used in the Entity Linking task of the TAC 2009 [71] are also widely adopted. The metrics include Micro-Averaged Accuracy which measures entity linking accuracy averaged over all the name mentions and Macro-Averaged Accuracy which measures entity linking accuracy averaged over all the target entities. In Table 2.9, we show examples of works that use the above metrics.

Table 2.9: Metrics usages

Method	Precison	Recall	F1	Micro-Averaged	Macro-Averaged
[25]	*	*	*		
[68]	*	*	*		
[136]				*	*
[95]				*	*
[44]				*	*
[85]			*		
[45]	*	*	*		

Gold Standard Datasets

Many manually annotated data sets are contributed by researchers and made publicly available. Therefore, these data sets are good benchmarks for evaluating an entity linking system. Some details of these data sets can be found in [21,42]. Moreover, a publicly available benchmarking framework for comparison of entity-annotation systems is recently proposed by Cornolti et al. [21]. Below are descriptions of a collection of popular datasets.

1. **AIDA-YAGO2 Dataset.** The AIDA-YAGO2 dataset [5] [50] is an extension of the CoNLL 2003 entity recognition task dataset [114] which is based on news articles published between Aug. 1998 to 1997 by Reuters. Each entity is identified by its YAGO2 entity name, Wikipedia URL, and Freebase if available.

2. **Microposts2014/2015 NEEL.** The 2014 Microposts dataset [6] [13] contains 3,504 tweets that are extracted from over 18 million tweets over one month in 2011. The 2015 corpus [98][7] includes more tweets (6,025) and covers more noteworthy events from 2011 to 2013.

3. **OKE2015.** The Open Knowledge Extraction Challenge 2015 [8] corpus contains 197 sentences obtained from Wikipedia articles. The mentions are linked to DBpedia.

4. **RSS-500-NIF-NER.** The RSS-500 dataset[9] [100] extracts data from 1,457 RSS feeds, including major international newspapers. It covers topics such as busi-

[5]https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/aida/downloads/
[6]http://scc-research.lancaster.ac.uk/workshops/microposts2014/challenge/index.html
[7]http://scc-research.lancaster.ac. uk/workshops/microposts2015/challenge/index.html
[8]https://github.com/anuzzolese/ oke-challenge
[9]https://github.com/AKSW/n3-collection

Table 2.10: Datasets usages

Dataset	Works
AIDA-YAGO2	[85], [129], [21]
Microposts 2014/2015	[99], [15]
OKE2015	-
RSS-500	[120], [111], [53]
WES2015	[20]
WikiNews	-
TAC-KBP 2009	[44], [45], [136]
Meij's	[39], [109], [68], [25]

ness, science, and world news. The RSS500 corpus contains 500 manually selected sentences and they were annotated by one researcher.

5. **WES2015.** The WES2015 dataset[10] [123] contains 331 documents annotated with DBpedia entities where these documents are from a blog[11] on the history of science, technology, and art. The dataset also includes 35 annotated queries inspired by the blog's query logs, and relevance assessments between queries and documents.

6. **WikiNews.** The WikiNews dataset is compiled by the NewsReader project [12] [121] which contains 120 Wikinews articles, classified into four sub-corpora: Airbus, Apple, General Motors and Stock Market. The articles are annotated with entities in DBpedia.

7. **TAC-KBP2009.** TAC-KBP2009 which is provided by The Knowledge Base population (KBP) track which is part of NIST Text Analysis Conference (TAC)[13]. In this dataset, there are 3904 entity mentions in total distributed in 3688 documents, each of which includes at most two mentions.

8. **Meij's** Meji's dataset is a famous public available dataset to evaluate tweet annotation which is contributed by Meji et al. [72], to create the dataset, they have asked two volunteers to manutally annotate 562 tweets, each containing 36.2 terms on average.

Moreover, for each gold standard dataset, we find works that use the dataset as shown in Table 2.10.

[10] http://yovisto.com/labs/wes2015/wes2015-dataset-nif.rdf
[11] http://blog.yovisto.com/
[12] http://www.newsreader-project.eu/results/data/wikinews
[13] http://www.nist.gov/tac/about/index.html

2.2.6 Summary

In this section, we presented a comprehensive study for entity linking. Specifically, we show the applications of entity linking and introduced various methods in two major steps in entity linking task that are 1) candidate entity generation and 2) candidate entity ranking. Also, we introduced the evaluation metrics and datasets used to evaluate an entity linking system.

Chapter 3

The Proposed Approaches

The first objective of our work is to develop a semantic relatedness measure between two words, regardless of whether they have explicit semantics (e.g., dictionary words) or have no formal semantics (e.g., hashtags or slang words), based on their occurrence on Twitter. The reason we want to design a semantic relatedness measure based on Twitter content is that existing work in the semantic relatedness literature has already considered various information sources such as WordNet, Wikipedia and Web search engines to identify the semantic relatedness between two words; however, such measures might not be directly applicable to microblogging content such as tweets due to 1) the informality and short length of microblogging content, which can lead to shift in the meaning of words when used in microblog posts, 2) the presence of non-dictionary words that have their semantics defined/evolved by the Twitter community. Therefore, we propose the Twitter Space Semantic Relatedness (TSSR) technique that relies on the latent relation hypothesis to measure semantic relatedness of words on Twitter. We construct a graph representation of terms in tweets and apply a random walk procedure to produce a stationary distribution for each word, which is the basis for relatedness calculation. Our experiments examine TSSR from three different perspectives and show that TSSR is better suited for Twitter analytics compared to the standard semantic relatedness techniques.

The second objective of our work is to create an entity linking technique targeted for Twitter content by applying semantic relatedness method we propose before. Entity linking (also known as semantic annotation) of textual content has received increasing attention. Recent works have focused on entity linking on text with special characteristics such as search queries and tweets. The semantic annotation of tweets is specially proven to be challenging given the informal nature of the writing and the short length of the text.

Therefore, we propose a method to perform entity linking on tweets built based on one primary hypothesis. We hypothesis that while there are formally many possible senses for an ambiguous term, as listed on the disambiguation page of their on Wikipedia, there are only few senses that are likely to be employed in the context of Twitter. Based on this hypothesis, we propose a method to identify such *dominant senses* for each ambiguous term and use them in the annotation process. Particularly, our proposed work integrates two phrases i) *dominant sense detection*, which applies community detection methods for finding dominant sense for ambiguous term; and ii) *tweet annotation* that links a tweet with entities in Wikipedia by only considering the identified dominant senses. We show that our proposed work offers competitive results with state-of-the-art methods while only considering a limited set of senses for ambiguous terms.

In the following subsections, we will introduce the semantic relatedness method and entity linking method in detail, respectively.

3.1 Semantic Relatedness Method

Figure 3.1 is the overall workflow of our proposed semantic relatedness method. As shown in the figure, we first retrieve a collection of tweets by querying Twitter REST API. We then process the raw tweets by removing the stop words, performing stemming on the words, and tokenizing the words. Afterwards, we calculate the relationships between each pair of words by counting the co-occurrences of these words, and can create a Twitter space graph whose nodes are words obtained from tweets, and the edges are conditional probabilities from one node to another based on the co-occurrence information. Finally, a random walk is performed on the graph to get a stationary distribution for each word, and cosine similarity method is used to compute the similarity between the two stationary distribution as the final score of semantic relatedness.

In the following, we show the details of the workflow. We first define the problem we are addressing by formally defining all relevant concepts, and then present the proposed method in detail.

First, we define a tweet since our work is especially designed for analysing Twitter content. By defining tweets, we can get a better understanding of the features of tweets and also generate statistics of users' posts.

Definition 3.1.1 (Tweet) A Tweet t is defined as a triple, $t = (userId, tweetId, body)$, where $t.userId$ is the unique Id associated with each Twitter user, $t.tweetId$ is a unique Id of each Tweet t, $t.body$ is the textual content of t. The $t.userId$ indicates the user who posts the underlying tweet, by aggregating the tweets with the same user Id, we

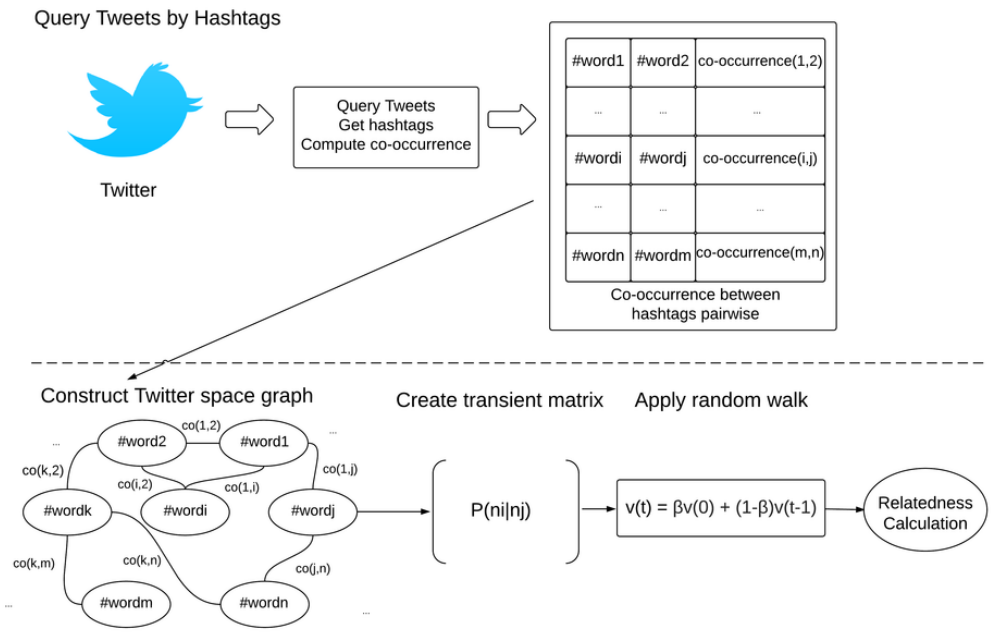

Figure 3.1: Semantic relatedness workflow

can analyze different users' posting styles and may use this feature for future work. The $t.tweetId$ is a unique identifier for a tweet, we can use that to fast search a tweet and create the index. Finally, the $t.body$ is a string of words which provides semantic information, by utilizing and generating statistics from the text string, we can explore the insight semantics behind it.

Based on the Tweet Definition 3.1.1, we can classify tweets according to a specific user u. We denote a set of Tweets that belong to a specific user as $T_u = \{t|t.userId = u\}$. We define the collection of all Tweets as T. Then, our graph will be generated from the collection of all Tweets.

In order to create a graph representation of terms in tweets to apply the random walk, we need to split the text content of a tweet into words. Therefore, we define a Tweet Token as following.

Definition 3.1.2 (Tweet Token) A Tweet Token tk is defined as a quadruple, $tk = (t, tokenId, token, isHashtag)$, where $tk.t$ corresponds to the underlying Tweet t, $tk.tokenId$ is a unique numeric identifier associated with each token, $tk.token$ is the stemmed form of the word in t. We include $tk.isHashtag$ to indicate if a token is hashtag or not, because hashtags in Twitter usually have specific format, e.g. abbreviation, new

invented slang which often do not have explicit semantics. Therefore, since we state that our work can identify the semantics of words regardless of whether they have explicit semantics or not, we can use hashtags in tweets to evaluate and validate our proposed methods.

Based on the definition 3.1.2, the collection of tokens[1] within a given tweet can be represented as $TW_t = \{tk.token|tk.t = t\}$. Furthermore, we denote the Tweet Words set as $TW = \{tk.token|tk.t \in T\}$, which is the collection of all tokens observed across all tweets. This collection of all tokens is then used to build the graph for later use.

So far, we have the collection of all tokens extracted from tweets to be the nodes in the graph, we need to find the relationships between nodes to present the edges in the graph. Therefore, we define co-occurrences between two tokens as below.

Definition 3.1.3 (Co-occurrence) Given two tokens w_i and w_j in TW, we define their co-occurrence count as $co(w_i, w_j) = |coT(w_i, w_j)|$ where $coT(w_i, w_j) = \{t|w_i \in TW_t and w_j \in TW_t\}$. In other words, if two tokens appear together in the same tweet, then their co-occurrences increment once.

Definition 3.1.3 will support our basic assumption that the more two tokens occur in the same tweet, the more related they are.

In order to apply the random walk algorithm, not only we need the relations between two nodes, but also we need to define the probability from one node to another. To do so, the conditional probability is defined to achieve the goal.

Definition 3.1.4 (Conditional Dependency) Given a token w_j and its co-occurrences with other tokens in TW, we define conditonal dependencey, $CD(w_i|w_j)$, as the probability of observing w_i if and when w_j is observed, which is calculated as follows:

$$CD(w_i|w_j) = \frac{co(w_i, co_j)}{\sum_{w_k \in TW} co(w_j, w_k)} \tag{3.1}$$

The conditional dependency defined in Definition 3.1.4 ensures that semantic relatedness is dependent not only on the co-occurrence of the two tokens together but also on the co-occurrence of each of the tokens with other tokens in the corpus. In other words, if a token has high co-occurrence with many tokens in the corpus, it is likely that this token is less specific and therefore should receive a lower degree of semantic relatedness.

The basic premise of our work is on the latent relation hypothesis [118] that states that pairs of words that co-occur in similar contexts tend to have similar semantics. Therefore, we hypothesize that the semantics of the words on Twitter can be derived from the context in which they appear, which is typically the tweets where those words

[1]From here onwards, the terms word and token are used interchangeably.

are observed. The rationale for choosing individual tweets as the context is that tweets often focus on a very specific subject and therefore each word is only used in one specific sense in a given tweet, even in the case of ambiguous words. For this reason, considering each tweet as the context allows us to focus on specific senses of each word.

Based on this, we build a co-occurrence graph in which the tokens that appear in the same contexts are connected to each other.

Based upon the above definitions, now we can create a graph to apply the random walk algorithm. We name the graph Twitter Word Dependency Graph and it is defined as follows.

Definition 3.1.5 (Twitter Word Dependency Graph) Given TW and $CD(w_i|w_j)$, we define Twitter Word Dependency Graph as a weighted directed graph, denoted as TWDG, where $w_i \in TW$ are the notes, and $CD(w_i|w_j)$ is the weight of teh edge from node w_j to w_i.

Based on TWDG, we model semantic relatedness as being the probability of reaching one token from the other based on a random walk on the graph. In other words, we employ a random walk model where a particle is assumed to float through TWDG starting from a certain token node. The probability of finding the particle at a certain node such as w_i after t iterations is equivalent to the sum of all paths through which the particle could have reached w_i starting from any other node at the time $t-1$; this is formalized as:

$$w_i^{(t)} = \sum_{w_j \in TW} w_j^{(t-1)} CD(w_i|w_j) \tag{3.2}$$

Now, given token w_j, the objective is to find a stationary distribution for it by releasing a particle into TWDG and iteratively applying the random walk process. The stationary distribution for w_j can be represented as the distribution of the probability of the particle being found in each of the nodes of the graph after the application of the random walk process. In order to compute the stationary distribution, we first define an initial distribution $v(w_j)^{(0)}$ that places all of the probability mass on a single token node. Then, at each iteration of the walk, the distribution is updated with parameter β as follows:

$$v(w_j)^{(t)} = \beta v(w_j)^{(0)} + (1-\beta) M v(w_j)^{(t-1)} \tag{3.3}$$

where M is the transition matrix corresponding to the TWDG graph denoting the conditional dependency $CD(w_i|w_j)$ moving from node w_j to w_i. Hughes and Ramage [52] have proposed that a random walk process is rather insensitive to the value of the

β parameter and have suggested that it can be set to 0.1. They have also empirically evaluated that $v(w_j)^{(t)}$ converges to its unique stationary distribution $v(w_j)^{\infty}$ after a number of iterations proportional to $\beta - 1$. For us, the convergence criteria was set to $|v(w_j)^{(t)} - v(w_j)^{(t-1)}| < 10^{-6}$ for which our experiments showed to converge in around 20 iterations.

Given the stationary distribution of each token derived from the random walk on TWDG, we measure the similarity of two tokens by calculating the similarity between their stationary distributions. As suggested in the literature [52], we use cosine similarity to measure the similarity of two tokens according to their distributions.

Definition 3.1.6 (TSSR) Semantic similarity of two tokens w_i and w_j in TSSR is defined based on the cosine similarity of their respective stationary distributions, $v(w_i)^{(\infty)}$ and $v(w_j)^{(\infty)}$ as follows:

$$SR(w_i, w_j) = \frac{v(w_i)^{\infty} \cdot v(w_j)^{(\infty)}}{|v(w_i)^{(\infty)}||v_(w_j)^{(\infty)}|} \tag{3.4}$$

As shown above, our proposed semantic relatedness system is built on Twitter content including the tokens we extracted from tweets and relationships between tokens according to the co-occurrences between tokens. Therefore, whether the token has explicit semantic meaning or not, we can still apply random walk starting from that token to get a stationary distribution for it, then we can use the stationary distribution to do further computation.

3.2 Entity Linking Method

In this section, we introduce the details of our proposed method for entity linking on Twitter which includes two major parts: 1) Dominant Sense Detection and 2) Tweet Annotation.

3.2.1 Dominant Sense Detection

To detect dominant senses of ambiguous terms from a collection of tweets $\mathbb{T}$, we follow two steps: *sense clustering* and *sense mapping*. Figure 3.2 shows the workflow of the dominant sense detection. As shown in the graph, a term dependency graph is constructed from tweets, the nodes are words extracted from tweets, the edges between nodes are eliminated from the figure due to limited space and the values of the edges are calculated based on the semantic relatedness method we proposed before. After that, a sense clustering process is performed by applying a clustering method on the term

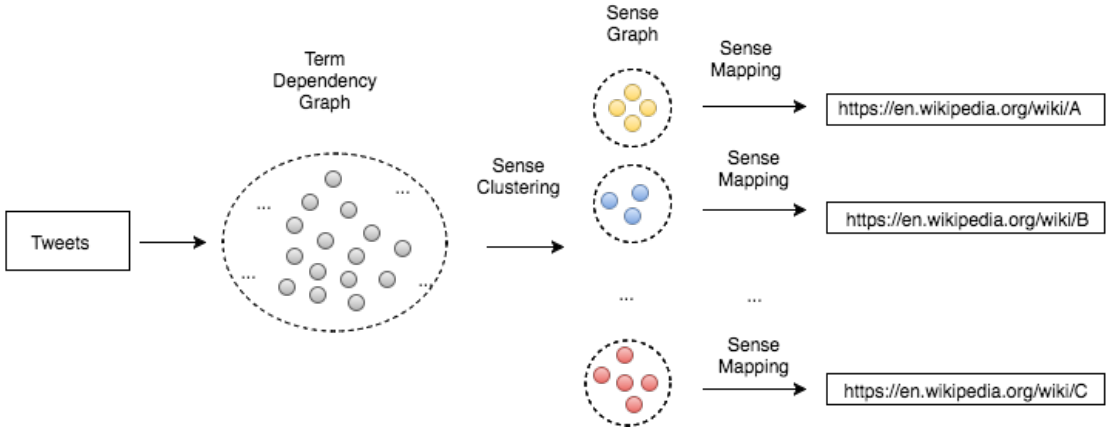

Figure 3.2: Dominant sense detection work flow

dependency graph to split the graph into subgraphs called sense graph as shown in the middle of the figure, each sense graph contains a set of words representing a sense. Finally, in order to have a valid representation of senses, we map each sense cluster to a valid Wikipedia article through a sense mapping process. The details of each step are described in the following sections.

Sense Clustering

The purpose of this step is to find the set of all possible dominant senses for a given ambiguous term as they appear on Twitter. To find the senses for a given ambiguous term, we first find the terms that are most similar and highly related on Twitter to this ambiguous term, and then cluster the identified terms based on their degree of similarity. The hypothesis is that the produced clusters would represent the senses of the ambiguous term as shown in Figure 1.2. Let us describe this process more formally.

To find all ambiguous terms $\mathbb{A}$ from $\mathbb{T}$, given a tweet $t \in \mathbb{T}$, we extract all possible n-grams from t. For each n-gram, we use Wikipedia API[2] to perform lexical matching on titles of Wikipedia articles in order to find its associated concept c from the Wikipedia concept set $\mathbb{C}$. An n-gram is considered to be ambiguous, $a \in \mathbb{A}$, if its corresponding Wikipedia concept c has a disambiguation page in Wikipedia. The remaining terms are placed in a disjoint set $\mathbb{W}$.

To extract dominant senses for the identified ambiguous terms $\mathbb{A}$, we create a graph, called the *term dependency graph*. The graph is built based on the *latent relation* hy-

[2] pypi.python.org/pypi/wikipedia/

pothesis that expresses that terms appearing in similar contexts carry similar or related semantics. In order to build the graph, we represent each term seen in our corpus as a vertex and their edges would denote normalized co-occurrence of the terms within tweets. Based on this graph, it is now possible to identify the terms that have been most frequently observed with an ambiguous term. More formally:

Definition 3.2.1.1 (Term Dependency Graph) A *term dependency graph* denoted as $\mathcal{G} = (\mathbb{V}, \mathbb{E}, g)$ is a weighted directed graph in which $\mathbb{V}$ includes all of the terms that have co-occurred with ambiguous terms, notationally, $\mathbb{V} = \mathbb{A} \cup \mathbb{W}$. $\mathbb{E}$ denotes a set of weighted edges e_{w_i,w_j} from term w_j to term w_i whose weight $g(e_{w_i,w_j})$ is calculated using the following conditional dependency between terms:

$$g(e_{w_i,w_j}) = \mathrm{P}(w_i|w_j) = \frac{f(w_i, w_j)}{\sum_{w_k \in V} f(w_j, w_k)} \tag{3.5}$$

where $f(w_i, w_j)$ is the number of times terms w_i and w_j have co-occurred in a similar tweet.

Now, given the term dependency graph $\mathcal{G}$, and an ambiguous term $a \in \mathbb{A}$, we apply the random walk algorithm [59] to find the most related terms to a denoted as $\mathbb{R}_a$, by starting the walk of a particle at the source node a. The probability of finding the particle at a certain node such as $w_j \in \mathbb{V}$ after l iterations is equivalent to the sum of all paths through which the particle could have reached w_j starting from any other node at iteration $l-1$. Formally;

$$w_j^{(l)} = \sum_{w_k \in \mathbb{V}} w_k^{(l-1)} P(w_j|w_k) \tag{3.6}$$

The stationary distribution for the target ambiguous term $a \in \mathbb{V}$ is obtained when the stationary distribution does not significantly change and can be defined as follows:

$$v(a)^{(l)} = \phi v(a)^{(0)} + (1-\phi) M_{\mathcal{G}} v(a)^{(l-1)} \tag{3.7}$$

where $v(a)^{(0)}$ is an initial distribution that places all of the probability mass on a single node, ϕ is the parameter to update the distribution at each iteration and $M_{\mathcal{G}}$ is the transition matrix associated with the term dependency graph $\mathcal{G}$.

Given the stationary distributions of terms, we can build $\mathbb{R}_a$ by ranking the terms in the graph according to the similarity of their stationary distributions and selecting the terms with the score higher than the average.

Once we have the most related terms for an ambiguous term, our next step is to identify different dominant senses for it. We consider the terms in $\mathbb{R}_a$ to represent all

the possible dominant senses of the ambiguous term a. In other words, this set of terms points to the most frequent senses for the ambiguous term. In order to separate and distinguish between the senses of the ambiguous term a, we first build a graph, called the sense graph, $\mathcal{SG}_a$, as follows:

Definition 3.2.1.2 (Sense Graph) a *sense graph* for an ambiguous term a, denoted as $\mathcal{SG}_a = (\mathbb{V}, \mathbb{E}, \gamma)$, is a weighted undirected graph in which $\mathbb{V}$ is the set of all highly related terms to a, i.e., $\mathbb{V} = \mathbb{R}_a$, $\mathbb{E}$ denotes a set of edges, and the weight function γ represents semantic relatedness between every two nodes in $\mathbb{V}$.

In our work, we compute the semantic relatedness between each two terms based on the method proposed by our previous work [29]. We adopt this semantic relatedness method instead of other existing state-of-the-art semantic relatedness methods because it is particularly designed for the Twitter sphere. In order to be able to identify the set of dominant senses of an ambiguous term from $\mathcal{SG}_a$, we would need to find separable clusters of this graph. To separate the senses, we focus on the fact that the terms about a certain *sense* are highly related to each other (maximal intra-sense term similarity) and terms from distinct senses do not share much relatedness (minimal inter-sense term similarity). For example, given an ambiguous term *apple*, one set of terms consists of {*fruit, juice, avocado, organic, leaf, banana*} while another includes {*model, watch, brand, factory, store, patent, jobs*}. While there is a high relatedness within each set, there is not too much similarity between the two sets. This implies that clusters within the sense graph could potentially represent the dominant senses of an ambiguous term.

To identify the dominant senses of an ambiguous term $a \in \mathbb{A}$, we apply clustering algorithms on $\mathcal{SG}_a$ to cluster the terms into distinctive senses. As a result, each ambiguous term a is associated with a set of dominant senses $\mathbb{S}_a$ each member s of which includes highly semantically coherent terms. In our experiments, we apply different clustering algorithms and compare their performance.

Sense Mapping

In order to employ dominant senses of an ambiguous term in the disambiguation process, we need to map each identified dominant sense to its appropriate Wikipedia concept. Denoting the Wikipedia concept set as $\mathbb{C}$, for each sense $s \in \mathbb{S}_a$ we consider all the corresponding concepts in the Wikipedia disambiguation page for the ambiguous term a as its candidate Wikipedia concepts, denoted as $\mathbb{C}_a \subseteq \mathbb{C}$. To find the best candidate concept c amongst $\mathbb{C}_a$ for s, we aggregate all of the terms in s as a single document and then calculate its similarity with the Wikipedia summary of each candidate concept in $\mathbb{C}_a$. The concept with the highest similarity is selected as the likely Wikipedia concept

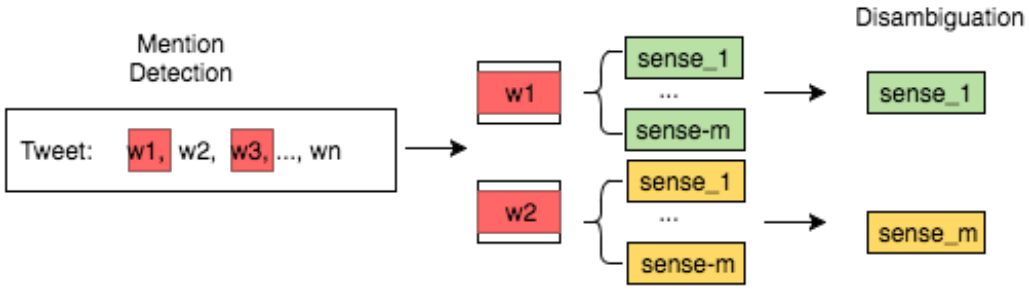

Figure 3.3: Tweet annotation workflow

for s. In order to calculate the similarity, we employ various similarity methods in our experiments and compare the results. In the sense mapping process, we may find the same Wikipedia concept for multiple senses of an ambiguous term. For example, as shown in Figure 1.2, for the ambiguous term *apple*, two senses are mapped to the same Wikipedia concept `Apple_(fruit)`. In such cases, we will merge the two or more dominant senses that were mapped to the same Wikipedia concept into one sense.

3.2.2 Tweet Annotation

The main objective of the annotation step is to link a short, noisy, and informal tweet to a set of Wikipedia concepts in order to provide a semantically coherent context for the tweet primarily based on the dominant senses identified from Twitter. In the tweet annotation task, we first identify mentions in the tweet that can be linked to a Wikipedia concept. Then, we associate the mention to an appropriate Wikipedia concept on-the-fly by utilizing dominant senses detected in dominant sense detection step, instead of all possible senses in Wikipedia. We explain this process through the *mention detection* and *disambiguation* steps. Figure 3.3 is the workflow of tweet annotation. First, given a tweet, by applying mention detection process, we get a list of potential candidate entities, such as w_1 and w_2 in the figure. If w_1 and w_2 are ambiguous, each of them would have a set of senses obtained from the previous section. Finally, the disambiguation process is performed to get the correct sense in the underlying scenario. The details of each step are presented below.

Mention Detection

In order to be able to detect the mentions in a given tweet t, we have built a *mention dictionary* by applying simple pattern matching on Wikipedia URLs to store all the recog-

nizable mentions. For example, given a Wikipedia URL presented as Apple,_Oklahoma, we consider the term *Apple* as a recognizable mention. We augment the mention dictionary by adding the titles of *redirect pages* and some variants of page title as suggested in [22]. Then, for each recognizable mention, we store the possible Wikipedia concepts plus the titles in its disambiguation pages.

We identify mentions of a given tweet t by performing lexical matching for all of the n-grams of t and checking them in the mention dictionary. The identified matches will be the list of mentions that we will annotate $\mathcal{M}_t$.

Disambiguation

Given the set of mentions $\mathcal{M}_t$ for tweet t, we link each mention $m \in \mathcal{M}_t$ to a Wikipedia concept $c \in \mathbb{C}$. In this process, if there is only one matching Wikipedia concept for a mention m, the mention is unambiguous and we directly link m to the corresponding Wikipedia concept c. Otherwise, if m belongs to the set of ambiguous terms $\mathbb{A}$, we consider its corresponding dominant senses $\mathbb{S}_m$ to be the candidates for disambiguation.

For example, for the tweet *'#NP Frankie Beverly and Maze Before I let go'*, there are two mentions, *Frankie Beverly* and *Maze*. Because there is only one possible Wikipedia concept for *Frankie Beverly*, i.e. Frankie_Beverly[3], we directly link it to this concept. However, the mention *Maze* is ambiguous and we consider two Wikipedia concepts Maze_(puzzle) and Maze_(band), which we have been identified as the dominant senses in the dominant sense detection phase for the term maze, as its candidates.

To associate an ambiguous mention $m \in \mathbb{A}$ to the best candidate from the set of its senses $\mathbb{S}_m$, we implement two similarity methods as follows.

Context Similarity: Based on the intuition that each annotation in a tweet should be related to the context of the tweet, we consider the similarity between each candidate $s \in \mathbb{S}_m$ and the target tweet t. To do so, we apply a document similarity method to calculate the similarity between t and the summary of the Wikipedia concept to which s is mapped, in the mapping step of dominant sense detection, as another document. The dominant sense with the highest similarity score will be selected as the annotation for that mention.

Collective Similarity: Similar to the previous works such as Kulkarni et al. [58] and Han et al. [45] that leverage the global coherence between candidate concepts, we apply collective similarity as defined in Equation 3.8 by considering both context similarity and the similarity between the candidate concepts with each other.

[3] https://en.wikipedia.org/wiki/Frankie_Beverly

Definition 3.2.2.1 (Collective Similarity) Given a set of k mentions for tweet t, $\mathcal{M}_t = \{m_1, m_2, ..., m_k\}$, and their dominant senses $\mathbb{S}_{m_1}, ..., \mathbb{S}_{m_k}$ as candidates of each mention, we let CP be the Cartesian product over k dominant senses, $CP = \mathbb{S}_{m_1} \times ... \times \mathbb{S}_{m_k}$. Collective similarity for each combination $CP_i \in CP$, $ColSim(CP_i)$ is calculated as follows:

$$\begin{aligned} ColSim(CP_i) = \prod_{j=1}^{|CP_i|} \prod_{k=j+1}^{|CP_i|} Sim(CP_i[j], CP_i[k]) \\ \times Sim(CP_i[j], t) \times Sim(CP_i[k], t) \end{aligned} \tag{3.8}$$

where $Sim()$ is a function that measures document-based similarity. Finally, we select the combination $CP_i \in CP$ with the highest score, $ColSim(CP_i)$, as the annotation set for the target tweet t.

3.3 Summary

In this section, we introduced the details of our proposed approaches including 1) semantic relatedness measurement and 2) entity linking, respectively. We first proposed a novel semantic relatedness method designed especially for twitter content. In order to do so, we created a Twitter Word Dependency Graph by extracting Tweet Tokens and Conditional Dependency between Tweet Tokens from the tweets, and then applied random walk algorithm to get a static distribution for each word as its representation. Afterwards, the cosine similarity method is employed on two static distributions to get the final semantic relatedness score between two words.

Second, we proposed an entity linking method by only considering dominant senses and used dominant senses to perform annotation. Our entity linking method consists of two steps that are 1) dominant sense detection and 2) tweet annotation. In the dominant sense detection step, we create a term dependency graph which presents the relationships between related terms of an ambiguous word, and apply clustering algorithms to obtain subgraphs from the term dependency graph and assume each subgraph to represent one sense of the ambiguous term. Next, by applying document similarity techniques, we map each subgraph to a valid unambiguous Wikipedia entry as its sense, therefore, we obtain a set of dominant senses for each ambiguous term. After that, in the tweet annotation process, we implement context similarity and collective similarity methodd to perform disambiguation and find the correct sense of each metion in a tweet.

Chapter 4

Empirical Evaluations

We conduct several evaluation experiments to show our work is effective and is competitive with the state-of-the-art. We introduce the datasets, gold standard, baselines, and metrics we adopt to evaluate our two works respectively.

4.1 Semantic Relatedness Method

We have benefited from the tweets dataset released by Cheng et al. [18] as the information source for building TWDG and for computing TSSR. After parsing the tweets and performing preprocessing such as removing stop-words and stemming the words in the dataset, we obtained 8,770,157 tweets with an average length of 8.4 words published by 106,349 users. These tweets were collected from 10 Nov 2006 until 17 March 2010. There were 4,148,886 unique words in total, which served as the vertices of the TWDG.

There are typically two approaches for evaluating the performance of semantic relatedness techniques. The first approach relies on a gold standard dataset of word pair similarities collected from a group of human subjects. The performance of a relatedness technique is measured through its degree of correlation with the subjective assessment of human subjects. The second approach evaluates two or more semantic relatedness techniques by measuring their impact on an application specific problem, e.g. their impact on improving product search. In this thesis, we evaluate TSSR using both approaches as well as a third strategy that consists of the subjective assessment of the ability of TSSR to describe frequent Twitter hashtags that do not have direct English language semantics. To sum up, we evaluate our work from three perspectives:

1. First, we employ the gold standard-based evaluation approach to compare the performance of our technique to the state of the art semantic relatedness techniques.

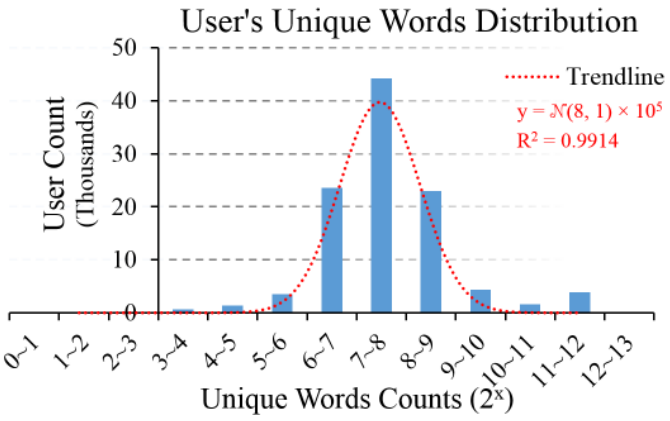

Figure 4.1: User's unique words distribution in the Twitter dataset

We benchmark our work against five other techniques from the literature on three different datasets.

2. Second, we implement and compare our work against the best performing semantic relatedness technique (ESA) on the application-specific problem of tweet search in order to observe how our technique performs in contrast to ESA.

3. Finally, given the fact that none of the existing semantic relatedness techniques is able to calculate the semantic relatedness of non-dictionary words, we perform an experiment involving human subjects to determine the suitability of our technique for semantically relating such words in practice.

In the following, we describe the details of our Twitter dataset and report on the three evaluation tasks.

4.1.1 Overview of the Twitter Dataset

As mentioned earlier, we used the Twitter dataset provided by Cheng et al. [18] that contains over 8.5M tweets and over 4M unique words. As shown in Figure 4.1, the majority of users used between 128 to 256 unique words across all the tweets in their timeline. There is a small number of users with a very small vocabulary, i.e., less than 64 unique words in total, or very large vocabulary, i.e., more than 512 unique words. This shows that the Twitter users that were covered in our dataset had a very focused and limited vocabulary that they frequently used. While we do not generalize this observation, we believe this might be a trend on Twitter since our dataset included over

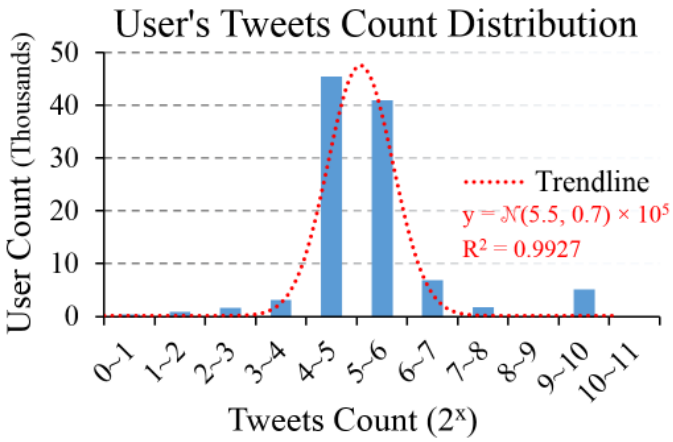

Figure 4.2: User's tweets count distribution in the Twitter dataset

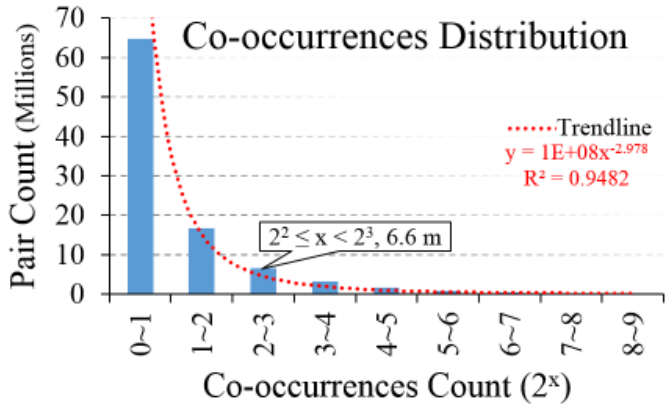

Figure 4.3: Co-occurrences distribution in the Twitter dataset

8.5M tweets. Furthermore, as shown in Figure 4.2, most of the users in our dataset posted between 16 to 32 tweets in the 3.5 year period. In terms of the co-occurrence of words in the Twitter data, a significant number of words had only been observed together once, which does not allow the derivation of any meaningful semantic relatedness between such words. Figure 4.3 shows the co-occurrence of words in the Twitter dataset. The co-occurrences are calculated by counting the number of times two stemmed words are seen together in the same tweet.

4.1.2 Gold Standard-based Evaluation

Traditionally, semantic relatedness techniques have been evaluated based on the correlation of their results with a gold standard dataset collected from human judges. These datasets include a collection of word pairs along with the assessment of human experts with regards to the similarity of the words in each word pair. For instance, WordSimilarity-353 collection contains 353 English word pairs [32], RG-65 consists of 65 word pairs [101] and MC-30 is a collection of 30 word pairs [82], which have been widely used in the literature. Many researchers [133] have used these datasets to show that their method is able to reasonably reproduce the word pair similarity rankings (not the actual relatedness value but the ranking of the word pair in the word pair dataset) by calculating spearmans rank correlation (ρ).

Table 4.1: Spearman's rank correlation and MAE results

	Spearman's Rank Correlation (ρ)			Mean Absolute Error (MAE)		
Method	WSW -353	RG -65	MC -30	WSW -353	RG -65	MC -30
ESA [33]	0.75	0.82	0.73	4.2	1.3	1.9
WikiRelate[1] [112]	0.49	0.52	0.45	-	-	-
Hughes and Ramage[1] [52]	0.47	0.76	0.84	-	-	-
WordNet-Res [97] [88]	0.30	0.55	0.72	4.3	1.5	1.5
WordNet-Path [48] [88]	0.19	0.50	0.48	3.6	1.7	2.1
TSSR	0.61	0.56	0.73	2.1	1.2	0.9

Our first assessment method consisted of benchmarking our work against the three aforementioned gold standard datasets and comparing the results with the state of the art semantic relatedness techniques [133] (see Section 4 for an overview of these techniques). The first three columns of Table 4.1 show the results of Spearmans rank correlation on the three datasets. As this table shows, on two of the datasets, TSSR does not perform as well as ESA. This is an expected result, consistent with the main hypothesis of our work: the semantics of words when used on Twitter differ from their more formal widely-used definition that has been employed in these datasets. However, even though the semantic relatedness derived by TSSR deviates from the one exposed by ESA, it is not overly remote from formal judgment rankings, as the computed rank correlations on the WSW-353 dataset demonstrate. Figure 4.4[2] clearly depicts the difference between

[1]Given we did not have access to the implementation of these two methods we resort to reporting results from [112] and [52]

[2]Due to space limitation, not all word pairs are listed on the x-axis.

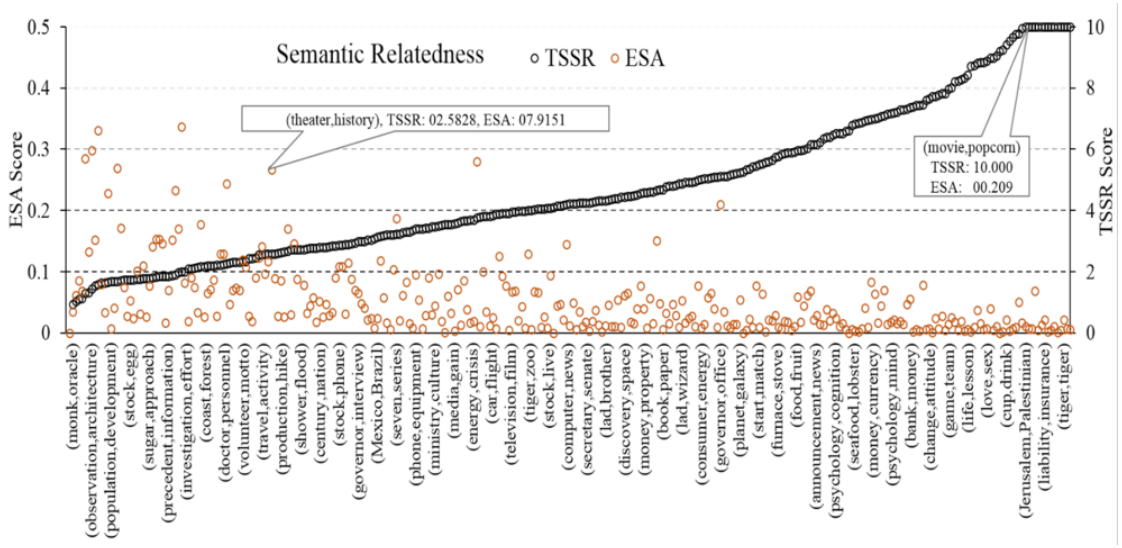

Figure 4.4: Comparison between ESA and TSSR semantic relatedness scores on the WSW-353 dataset

the semantic relatedness score distributions produced by TSSR compared to ESA. As an example, the figure shows that word pairs such as *game* and *victory* are not considered to be too highly semantically related in ESA or the WSW-353 dataset but are considered to be highly related by TSSR due to their frequent co-occurrence on Twitter. We will show in our next two experiments that such differences are a desirable effect of capturing the semantics of words based on Twitter context.

Apart from Spearmans rank correlation, in Table 4.1 we also report on the Mean Absolute Error (MAE) of the estimated semantic relatedness values produced by each method and the actual human judgments:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|m_i - g_i| \tag{4.1}$$

where n is the number of word pairs, m_i is the score produced by a semantic relatedness method for word pair i and g_i is the gold standard score for the same pair. In order to calculate MAE, the semantic relatedness values produced by different methods were scaled to [0,10], which is the scale used in gold standard datasets. As shown in Table 4.1, TSSR produces the smallest mean absolute error across all of the three gold standard datasets. This means that the value proposed by TSSR in the range of [0, 10] for each pair of words is closer to the actual value attached by human subjects compared

to other methods. However, statistically speaking, as observed in Table 4.1, the lowest MAE does not result in the highest rank correlation. In other words, a method can have a low MAE but produce a ranking that is not the same as the gold standard. It should be noted that the implementation of methods [112] and [52] were not publicly available; therefore, we were not able to generate MAE values for these two methods.

4.1.3 Tweet Search

The second evaluation strategy that we adopted was an application-based method. Given the fact that one of the most important application areas of semantic relatedness techniques is to improve search, we compared TSSR to ESA, which showed the best performance in the first experiment, when applied to the domain of tweet search. In order to integrate semantic relatedness into tweet search, we extended the baseline vector-based comparison of query terms with tweet space terms. Hence, the similarity of a tweet to a query is calculated as the sum of semantic relatedness between query terms and tweet terms as follows:

$$S_{tweet}(T,Q) = \sum_{i=1..k} \sum_{j=1..n} SR(q_i, w_j) \tag{4.2}$$

where n is the number of words in a tweet (T) and w_j is the j^{th} word in the tweet; k is the number of words in the query (Q) and q_i is the i^{th} word in the search query. For a given tweet T and a query Q, $S_{tweet}(T,Q)$ calculates the semantic relatedness between T and Q. In the search process, tweets are ranked based on their degree of semantic relatedness to the input query. We used TSSR and ESA for $SR(q_i, w_j)$ and performed our evaluation as follows.

For a given single-term query, we find 100 tweets that have the exact query term in their content; we refer to these as target tweets. We then identify 900 tweets that neither contain the exact query term nor have topical similarity with the query term (determined by the human expert); we refer to these as irrelevant tweets. We then anonymize the target tweets by removing the exact query term from the tweet content. Therefore, the overall dataset of 1,000 tweets contains 10% of relevant tweets and 90% irrelevant tweets, none of which have the exact query term in them. The objective is to study whether and to what extent the search method based on $S_{tweet}(T,Q)$ is able to find the target tweets based on the computed semantic relatedness values and without the presence of the exact query term.

For the purpose of experimentation, we selected the 100 most frequent words in the overall dataset used in this study, as the 100 queries to be used for search. We

performed the above procedure for each of the 100 queries and employed the standard TREC evaluation tool to compute the performance measures. We report three metrics in Table 4.2, namely i) Mean Average Precision (MAP), which is the mean of the average precision scores of each query; ii) Reciprocal Rank that shows the multiplicative inverse of the rank of the first correct answer; and iii) Precision at 100 (P@100), which shows the ratio of correct tweets in the top 100 results.

Table 4.2: Semantic search over tweets

Method	**MAP**	**Reciprocal Rank**	**P@100**
ESA	0.31	0.79	0.33
TSSR	0.39	0.95	0.38

The results reported in Table 4.2 show that TSSR is more effective in finding a higher number of tweets from the target tweet set. Given the search for relevant tweets in this evaluation strategy is only dependent on the performance of the semantic relatedness technique, we believe that the results are an indication that the semantic relatedness derived by TSSR for word pairs on Twitter is more accurate and representative of the semantics of words as they are used by Twitter users. Therefore, as observed in Figure 4.4, a shift can be observed between the semantics of a word on Twitter and it's common semantics, which if captured as done in TSSR, can lead to a higher performance when performing tweet search and possibly other Twitter related applications.

4.1.4 Describing Hashtags

The third evaluation strategy was to determine whether our semantic relatedness technique is able to identify the correct semantics of words that do not necessarily have explicit English language semantics such as hashtags. We performed this evaluation with 35 human participants, all with a good understanding of the Twitter dynamics. We have randomly selected the participants; therefore, the selection may not be a fair representative of typical Twitter users. Each participant was given a hashtag along with a set of 25 descriptive words that described that hashtag. The descriptive words for a hashtag were derived by using TSSR to find the top 25 words that had the highest semantic relatedness with that hashtag. Table 4.3 shows five sample hashtags and their descriptive words that were included in the experiments. Each participant was then asked to provide their perspective on the following three statements regarding the relationship between the hashtag and its 25 descriptive words:

1. *The 25 words for the given hashtag are highly descriptive.*

Table 4.3: Sample hashtags and their descriptive words set

Hashtag	**The 25 Descriptive Word Set**	**Meaning**
HW	finish, done, home, class, school, help, hour, math, read, study, tired, book, assign, write, paper, idea, problem, pic, homework, stupid, page, monday, spanish, teacher	Homework
MJS	jackson, show, michael, song, movie, miss, music, die, listen, world, dance, memory, death, fan, perform, hear, beat, remember, white, gone, sing, left, album	Michael Jackson
TCOT	tlot, p2, obama, gop, sgp, teaparty, health, news, care, bill, show, healthcare, hcr, vote, video, palin, ocra, help, job, conservation, read, talk, president, plan	Top Conservatives on Twitter
STEM	research, study, current, grow, health, world, education, institution, school, derive, approve, challenge, science, stress, success, conflict, learn, brain, technology, develop, vote, university, student, support	Science, Technology, Engineering, and Math
BOHO	style, chick, bag, leather, brown, hair, shop, tan, tote, fashion, saddle, wooden, trendy, show, wear, sale, rock, pretty, design, dress, store, cut, jacket, shoe	Informal and Unconventional Fashion

2. *There are no irrelevant descriptors within the 25 words set.*

3. *There are no important missing descriptors from the 25 words set.*

The participants were asked to provide their assessment using a five level Likert-type scale. The purpose of the first statement was to determine whether the correct semantics of the hashtag was identified by TSSR. The second statement focused on an informal assessment of precision, while the third statement evaluated perceived recall. We selected the top 50 most frequent hashtags in our dataset and each hashtag was independently assessed by seven participants.

First, in order to examine whether the opinions provided by human participants were consistent and that valid conclusions can be drawn from the data, we performed inter-rater reliability analysis. In particular, we applied intra-class correlation (ICC), which is a descriptive statistic that is used to measure the agreement within a group of individuals. In Table 4.4 we report both ICC single measure and ICC average measure. The former defines the extent to which the opinion of a single participant is similar with the other

Table 4.4: Intra-class correlation (ICC) of the participants

ICC single measure	ICC average measure
0.757	0.956

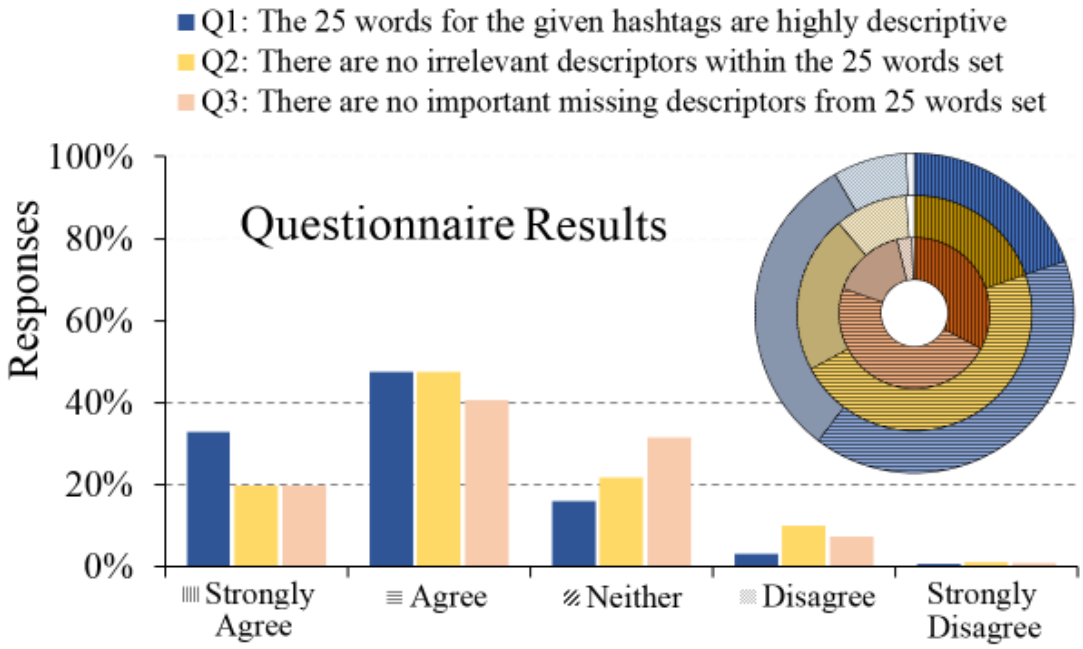

Figure 4.5: Results of the hashtags study

participants, whereas the latter shows how reliable it is to use the average opinions of participants. As shown in Table 4.4, the ICC single measurement value (0.757) shows a reasonable agreement among the participants, while the ICC average result (0.956) shows the reliability of the study.

Given that we have demonstrated that the participants were highly consistent in their responses to the three questions (high ICC values); therefore, it is reliable to use the median of the values received for each of the questions to represent the subjective opinion of the participants. The median of the answers for all three questions was 4, which corresponds to *agree* (5 for strongly agree and 0 for strongly disagree), which is an indication that the participants collectively agreed with the three statements regarding descriptiveness, precision and recall of the 25 descriptive words set extracted by TSSR for each of the hashtags. This shows the fact that TSSR has been able to identify the semantics of the hashtags, i.e., words that do not have explicit English language semantics, with a reasonably high quality. The distribution of the answers received from

the participants is shown in Figure 4.5.

Threats to validity

In this section, we analyze the threats to the validity of our experiment with human subjects.

The validity of an empirical experiment can be affected by different threats. Therefore, we will discuss the threats to conclusion, construct, internal and external validity. We will also specify the aspects of our experiments that may have been affected by these threats.

Conclusion Validity: Conclusion validity is the extent to which the conclusion of a relationship between the treatments and the results are valid. A threat to conclusion validity is a factor that can lead a researcher to reach an incorrect conclusion. There are two types of errors: 1) The conclusion that there is not relationship when in fact there is and 2) The conclusion that there is a relationship when in fact there is not. In our experiments, a limited number of hashtags was collected with their 25 descriptive words. Although 25 words can express the meaning of a hashtag, but increasing or decreasing the number of descriptive words we select may affect the decision. Also, a limited number of participants were randomly selected to join the experiment. For instance, if a participant uses Twitter on a daily basis, he/she can understand the hashtag more easily, therefore, a participant could impact the results of the experiment. In future work, we can also study the background of each participant and select different number of descriptive words to replicate the studies.

Construct Validity: Construct validity has traditionally been defined as the experimental demonstration that a test is measuring the construct it claims to be measuring. In our experiments and based on the statement we claim, we have been interested in measuring and analyzing the ability of our semantic relatedness method to recognize English words that do not have explicit meanings. Therefore, we create a set of descriptive words for each target word and present it to participants and ask questions regarding the understanding of the word. One of the threats that could be attributed to this measurement approach is that some participants in our experiments might have prior knowledge of these words and that would indirectly influence their understanding of the words. Therefore, in future work, we can study the knowledge of the participants of these target words.

Internal Validity: Internal validity is the extent to which researchers are able to say that no other variables except the one researchers are studying caused the result. There are several common threats to internal validity. For example, one threat to internal

validity is selection. In our experiment, this is simply the fact that the people who are selected in the study may not be general. As mentioned earlier, we randomly select participants, however, they do not represent all Twitter users. Therefore, they may not represent the general Twitter users' understanding of the experiments. Another threat to internal validity is maturation. How do we know that people would not change during the study because they matured. For example, if a participant starts using Twitter more often than beginning of the study, the answer of the participant may differ as time goes by. Therefore, in future work, we can replicate the experiments in different time periods.

External Validity: External validity is the extent to which results of a study can be generalized to the world at large. The results of a completely externally valid study can be generalized and applied safely to the semantic understanding of words. In our experiments, the external validity can be firstly, we randomly select a collection of target words without explicit English meanings and provide 25 descriptive words to explain the meanings. Also the participants do not particularly need a high level of Twitter experience to be able to complete the experiment.

4.2 Entity Linking Method

4.2.1 Twitter Corpus

In order to find the dominant senses for ambiguous terms, we use the publicly available Twitter dataset[3] released by Cheng et al. [18] as our corpus. It consists of approximately 8 million tweets posted by 106,349 unique users between Nov 10, 2006 and March 17, 2010. In this corpus, the average number of terms in each tweet is 8.4 and there are 4 million unique terms available in the corpus.

4.2.2 Gold Standard

We adopt the Twitter dataset published by Meij et al. [72] as the gold standard to evaluate our annotations. It consists of 502 tweets which are manually annotated by two volunteers. In this dataset, the average number of annotations for each tweet is 2.17 and there are 127 tweets that do not have any annotations. There are two other datasets released by Tran et al. [115] and Shen et al. [109]. However, we were not able to use them because they are limited only to annotations for trending hashtags and annotations customized to specific users and are not publicly available.

[3] https://archive.org/details/twitter_cikm_2010

Table 4.5: Results based on different parameters combinations

Clustering Method	Mapping Method	Annotation Method	Precision	Recall	F1
Louvain	WordsMatch	Context-based Similarity	**0.765**	**0.581**	**0.660**
		Collective Similarity	0.739	0.547	0.629
	UMBC Phrase	Context-based Similarity	0.728	0.530	0.613
		Collective Similarity	0.683	0.50	0.577
	UMBC STS0	Context-based Similarity	0.723	0.530	0.612
		Collective Similarity	0.675	0.489	0.567
K-means	WordsMatch	Context-based Similarity	0.744	0.563	0.641
		Collective Similarity	0.723	0.542	0.620
	UMBC Phrase	Context-based Similarity	0.725	0.526	0.610
		Collective Similarity	0.688	0.494	0.575
	UMBC STS0	Context-based Similarity	0.712	0.523	0.603
		Collective Similarity	0.677	0.500	0.575
Hierarchical	WordsMatch	Context-based Similarity	0.731	0.542	0.622
		Collective Similarity	0.715	0.533	0.611
	UMBC Phrase	Context-based Similarity	0.693	0.510	0.588
		Collective Similarity	0.683	0.490	0.571
	UMBC STS0	Context-based Similarity	0.712	0.522	0.602
		Collective Similarity	0.667	0.499	0.571

4.2.3 Metrics

Given the gold standard dataset, we adopt the evaluation metrics that have been used in the related literature [30, 51, 68, 130] to evaluate the quality of our work. We determine the quality of the annotations using standard information retrieval metrics including Precision, Recall and F-measure and compare the performance of our proposed method with other state-of-the-art benchmarks.

4.2.4 Experimental Setup

There are three main variation points in our proposed approach, which can affect the performance of our results:

The choice of the clustering method. As mentioned in Section 3.1.1, we require a clustering method to detect dominant senses of an ambiguous term. We select three different clustering algorithms, namely *Louvain* [6], *K-Means* [2] and *Agglomerative* hierarchical clustering [104].

Louvain [6] is an efficient heuristic method that finds clusters by optimizing both *modularity* and extraction time on a weighted graph. The *K-Means* method [2] is a widely used clustering technique that aims to minimize the average squared distance between

points in the same cluster and maximize inter-cluster dissimilarity. The *Agglomerative* clustering method [104] performs hierarchical clustering using a bottom-up approach and builds nested clusters by merging or splitting them successively.

It should be noted that *Louvain* does not require a priori knowledge of the number of clusters (β) when running the algorithm and β is determined by the algorithm itself. However, the other two algorithms require the number of clusters to be predefined. Therefore, we apply β obtained from Louvain as the number of clusters in the other two methods *K-Means* and *Agglomerative*. The results are reported in Table 4.5. In addition to using β as the number of clusters in these two methods, we also evaluate larger and smaller cluster sizes around β for these two methods. According to our experimental results, we observed that varying the number of clusters around β does not lead to any meaningful improvements in our final results in either *K-Means* or *Agglomerative* clustering. Therefore, we do not report these results in Table 4.5.

The choice of the similarity method in sense mapping. As mentioned in Section 3.1.2, we apply a document similarity method to map a dominant sense which is represented as a set of terms to a Wikipedia concept. To do so, we adopt three state-of-the-art document similarity methods: *i*) *Words Match Similarity* [36]; *ii*) *UMBC Phrase Similarity*[4] and *iii*) *UMBC Semantic Textual Similarity*[5]. The document similarity methods proposed by UMBC [43] are based on distributional similarity and Latent Semantic Analysis (LSA) [27] combined with semantic relations extracted from WordNet[6] and they assume the semantics of a phrase/text is dependent on its component words.

The choice of disambiguation method. As introduced in Section 3.2.2, we implement two disambiguation methods, namely *Context Similarity* and *Collective Similarity*. The choice of the disambiguation method can impact the performance of the annotation process. We report our experimental results for both of the disambiguation methods.

By selecting and combining the different alternatives for these three variation points, we obtain 18 variants (3 clustering techniques $\times$ 3 document clustering methods $\times$ 2 disambiguation techniques) that are evaluated and compared using the gold standard dataset in terms of Precision, Recall and F-measure. The results are shown in Table 4.5. By fixing two of the variation points, i.e., the clustering and mapping methods, we can compare different annotation methods. Based on Table 4.5, Context Similarity performs better than Collective Similarity in our work in terms of all the three

[4] http://swoogle.umbc.edu/SimService/phrase_similarity.html
[5] http://swoogle.umbc.edu/StsService/index.html
[6] https://wordnet.princeton.edu/

evaluation metrics. For instance, if we select the *Louvain* clustering method and the *words match* mapping method, the precision, recall, f-measure for *context similarity* is $0.765, 0.581, 0.660$, respectively while the same variant but with a *collective similarity* results in a lower performance of $0.739, 0.547, 0.629$. Given several researchers [45, 58] have mentioned that collective similarity performs better than other methods, we looked further for the reason why our observation was to the contrary. Based on our observations, we found that some Twitter users cram multiple pieces of information into one short-length tweet or there are tweets that cover multiple aspects that can mislead a collective similarity approach. Let us consider the following tweet: *'Dad doing his best charlie sheen impression. WINNINGGGGGG.'* In this tweet, when a collective disambiguation approach is used the term *Dad* is linked to the `Dad_(Angel)` concept in order to collectively disambiguate it with *Charlie Sheen*. In this case `Dad_(Angel)` is more similar to `charlie_sheen` compared to the correct sense which is *Dad*. There are many similar cases that are observed in tweets that will mislead a collective similarity approach and hence result in its poorer performance. Based on this we select context similarity as the choice for the disambiguation technique.

Similarly, we can compare the three mapping methods with each other. By fixing the clustering method and the annotation method, we observe that the *Words Match Similarity* mapping method produces higher results. By comparing the three clustering methods, it can be observed that using the *Louvain* clustering method results in higher quality annotations in terms of the three evaluation metrics. Therefore, we select the variant with the best performance to be compared with the state-of-the-art baselines, i.e., the variant composed of Louvain clustering, words match mapping and context similarity.

4.2.5 Comparison with Baseline Methods

In this section we first introduce the baseline methods and then we compare the quality and efficiency of our proposed method with the baselines.

Baseline Methods

The baselines selected for comparison can be divided into three categories: *i*) supervised, *ii*) semi-supervised and *iii*) unsupervised methods. Baselines belonging to the first category include Rysann [23], Liu et al. [68], Wikify! [75] and Meij et al. [72]. Rysann utilizes a probabilistic model that relies on a hybrid gaussian-hypergeometric combination to resolve ambiguities by producing the statistics on the distribution of words within

Table 4.6: Results for the set of baselines compared with our result

	Method	Precision	Recall	F1
Supervised	Rysann	0.752	0.595	0.664
	Liu's Method[7]	0.752	0.675	0.711
	Wikify![7]	0.375	0.421	0.396
	Meij's Method[7]	0.734	0.632	0.679
Semi-supervised	Huang's Method[7]	0.658	0.419	0.512
Unsupervised	TagMe	0.776	0.60	0.677
	Spotlight	0.621	0.453	0.524
	Our Method	0.765	0.581	0.660

each DBpedia concept, then a supervised training process is required to determine the scaling factors. Liu et al. [68] combine three types of local, entity similarity and mention similarity features. In order to combine these three types of features, they require a training process to determine the weight for each feature type. Wikify! [75] uses a combination of *knowledge-based* and *data-driven* methods and measures agreement by using a voting schema to perform disambiguation. The *knowledge-based* method is based on the overlap between the context of the potential concept and the keywords mentioned in the input text and the *data-driven* method uses a Naive Bayes classifier to integrate both local and topical features. Meij et al. [72] employ machine learning algorithms to focus mainly on the effectiveness of semantic linking as opposed to efficiency. As mentioned earlier, Huang et al. [51] are the only work that benefits from a semi-supervised approach where a smaller set of labeled data is required for their method. The main difference between our method and the above supervised methods is that our method is unsupervised which does not require labeled data for training and only considers the dominant senses in its disambiguation phase.

Unsupervised methods selected as our baselines include TagMe [30] which processes Wikipedia *anchor texts* and *pages* to cross reference mentions with Wikipedia articles, and DBpedia Spotlight [24], which builds a generative probabilistic model by processing the Wikipedia links with their anchor texts and textual context. Above mentioned unsupervised methods use external knowledge resources such as Wikipedia to obtain senses for the purpose of disambiguation, however, in our work, we generate dominant senses by only mining information from Twitter and only employ the identified dominant senses to perform disambiguation.

[7]Given we did not have access to the implementation of these four methods we resort to reporting results from [68], [75], [72], and [51]

Comparison Based on Gold Standard

The results of our comparison with the state-of-the-art baselines on the gold standard are reported in Table 4.7. To produce the results of the baselines, for Rysann[8], TagMe[9] and Spotlight[10], we used their RESTful API to annotate the tweets of the gold standard dataset. As for other baselines such as Liu's Method [68] and Huang's Method [51], we report their results obtained on the same gold standard dataset as reported in their papers. We opted for this method as the code for these works was not publicly available. With regards to the results for the method proposed by Meij et al. [72] and Wikify! [75], we employ the results reported in [68] which uses the same gold standard dataset, as the code for these techniques were also unavailable.

As shown in Table 4.7, while most of the state-of-the-art perform well on the gold standard but they employ a supervised strategy, which requires sufficient high quality labeled data in practice. Within the supervised method category, the best results in terms of all metrics are obtained by Liu's Method [68]. In their work, they consider not only local features, but also global features related to entity similarity and mention similarity and the results indicate the effectiveness of collective inference and global features. TagMe method performs the best in the unsupervised category. TagMe processes all Wikipedia pages which results in 3M anchors, 2.7M pages with a link-graph of about 147M edges, and computes a score for each possible sense from Wikipedia for a mention in order to perform disambiguation. As reported in the results, while TagMe shows the best performance overall, the performance of our approach is highly competitive with TagMe in all three metrics, i.e. precision (0.776 vs 0.765), recall (0.60 vs 0.581) and f-measure (0.677 vs 0.66). This can be viewed as a notable achievement when considering the fact that we only processed the dominant senses of ambiguous terms obtained from an 8M tweet corpus. Its interesting to note that according to the ambiguous terms that were present in the gold standard, the average number of senses defined on Wikipedia is 28 senses while we reduce this number to 5 based on the dominant senses that were identified.

One of the concerns that needed to be further investigated was whether the errors or omissions by our proposed method were due to the senses being incorrectly omitted when dominant senses were detected or not. In order to understand the source for the annotation errors or omissions that were made by our proposed technique, we manually reviewed all of the annotations that were generated against the gold standard and clas-

[8] denote.rnet.ryerson.ca/rysann
[9] tagme.di.unipi.it/tagme_help.html
[10] github.com/dbpedia-spotlight/dbpedia-spotlight/wiki/Web-service

Table 4.7: Comparative analysis of performance based on random sampled tweets.

Method	Precision	Recall	F1
TagMe	0.707	0.578	0.636
Spotlight	0.525	0.342	0.414
Rysann	0.717	0.519	0.602
Our Method	0.788	0.626	0.698

sified the errors and omissions into two categories: 1) errors or omissions that happened due to a missing sense eliminated in the dominant sense detection process, and 2) errors or omissions due to incorrect disambiguation. We found that in total and out of the 327 erroneous or missing annotations, only 75 ($\sim 22\%$) were due to the exclusion of the correct sense in the dominant sense detection process. This is a significant observation, which shows that the dominant senses provide a reasonably high coverage of the *right* senses that are needed in the tweet annotation process.

Comparison Based on Random Sampled Tweets

It is worth noting that we identified the dominant senses that were used in our experiments from a corpus of only 8M tweets. Given the limited size of our Twitter corpus, it is possible that some of the ambiguous terms in the gold standard were not observed in the Twitter corpus at all, which could have impacted our performance. As an example, for the tweet *'@yosoyjuanson are you REALLY in tasmania?? go to the MONA MUSEUM!! email me & i'll tell you who to talk to there!!!'* from the gold standard dataset, the term *tasmania* did not exist in our Twitter Corpus, therefore, we were not able to either identify the mention or detect any of its senses. This impacted the performance of our model in terms of *recall* reported in the previous section.

Furthermore, the basic hypothesis of our work is that a tweet should be annotated based on the dominant senses of ambiguous terms on Twitter. This hypothesis implicitly carries the fact that dominant senses of terms can change based on time. Therefore, ambiguous terms within a tweet would need to be annotated with dominant senses detected within the time period when the tweet was posted. However, given the fact that we were interested to compare with the state-of-the-art gold standard, there may have been temporal mis-alignment between the tweets in the gold standard and our Twitter corpus that could have affected the *precision* of our work. The best performance of our work will be achieved when the corpus and the tweets that are being annotated belong to the same time period and hence there is alignment between the dominant senses and the tweets. For instance, for the term *Apple*, in our Twitter corpus, we only found

Table 4.8: The mean and standard deviation of the execution times (in seconds).

Method	Mean	STDev
TagMe	0.005	0.019
Spotlight	0.0511	0.3910
Rysann	0.2888	0.6476
Our Method	0.001	0.001

Apple_(fruit) and **Apple corporation** as the dominant senses; however, it is possible that within a different time frame when the musician *Fiona Apple* is releasing a new album that the concept `Fiona_Apple` may turn out to be a part of the dominant sense as well. In order to show that temporal alignment matters in our approach, we created a second benchmark dataset that shares the same temporal alignment with our Twitter corpus.

To create the benchmark dataset, we adopted the approach proposed in Tuan et al. [115] and randomly sampled 100 tweets from our Twitter corpus. The sampled tweets were selected such that they each had at least five English words and included at least one ambiguous term. Two of the authors then carefully annotated each tweet before they were run through any of the annotators to create the gold standard.

We compare our method with TagMe, Spotlight and Rysann whose implementations are publicly available and report the results in Table 4.8. While the implementations for the other methods reported in Table 4.7 were not available for comparison, comparison with TagMe was considered to be a good indication of performance as it had one of the best performances on the previous gold standard. The results show that if the tweets are annotated based on the dominant senses detected from a temporally-aligned Twitter corpus that our method outperforms other state-of-the-art techniques (both supervised and unsupervised methods) in terms of precision, recall and f-measure.

In summary, our proposed work is an unsupervised method that generates dominant senses from the context of Twitter without relying on all senses from other knowledge bases and yet produces results that are competitive with the state-of-the-art. Based on the observed performance and comparison with the state-of-the-art, we conclude that we can positively respond to our first issue and conclude that the consideration of the dominant senses for ambiguous terms on Twitter can positively enhance the semantic annotation of tweets.

Execution Time Performance

In this section, we are interested in addressing our second issue as to whether 'the consideration of only a limited set of senses significantly reduces the annotation process time of tweets?' To this end, we compare the execution time of the different baselines for annotating the tweets in the primary gold standard. The experiments were conducted on an Intel(R) Xeon(R) 3.50GHz with 30GB RAM. We first deploy the baseline methods, whose implementations were publicly available, on our server and then calculate their execution time for annotating each tweet in seconds. The Mean and Standard Deviation (STDev) of the results for each method are shown in Table 4.8. Based on the results, our proposed method is the most efficient in terms of execution time, which is primarily due to two reasons: i) it only considers a small set of senses for the purpose of disambiguation and ii) it only uses context similarity, which is much less time consuming compared to collective similarity. In order to determine the statistical significance of the results, we ran a *paired t-test* between the execution times reported by our method for each tweet compared to TagMe, which is the next fastest approach. We obtained a p-value of <0.01, which shows statistically significant difference between the execution time of our approach compared to TagMe.

Based on these results, it is possible to address the both issues simultaneously that by relying only on dominant senses for the purpose of disambiguation, the entity linking process can be performed significantly faster while maintaining a competitive (or even better in the case of aligned corpus) performance in terms of precision and recall.

In order to better illustrate the performance of our work, we provide three examples: Given a tweet *'Sears 4Q earnings fall, adj. results top Street'*, the mention detected is *sears* which is ambiguous according to its Wikipedia *disambiguation page* and it has 17 senses defined on Wikipedia. However, we detect only six dominant senses for this ambiguous term including `Freddie_Sears`, `Sears_plc`, `Sears`, `Willis_Tower`, `Sears_Holdings_Corporation` and `Francis_Sears` and within these six dominant senses, our approach accurately selects the `Sears` concept as the correct annotation in this tweet. Another example is the tweet *'Tune into #msnbc for live cvg of Obama mini-press avail in Chile. About 2 mins away. #libya'*, one of the mentions detected is *Obama* and it has 17 concepts according to its corresponding Wikipedia *disambiguation page* and most of the concepts are rare in daily conversations such as `Mount_Obama` which is the highest point in Antigua and Barbuda or `Obama_Line` which is a railroad line operated by West Japan Railway Company. Instead of processing all the concepts defined in Wikipedia, our method detects two dominant senses for *Obama* that are `Barack_Obama` and `Obamacare` and by only considering these two dominant senses, our method successfully finds the

correct concept for the ambiguous term *Obama* in this tweet that is Barack_Obama.

Let us consider one further example and compare the performance of TagMe and our proposed approach. In a tweet *'Stay up Hawk Fans. We are going through a slump, but we have to stay positive. Go Hawks!'*, one of the ambiguous terms is *slump*. There are 11 disambiguation entries (senses) for this term on Wikipedia. When annotating this tweet with TagMe, the sense identified for *slump* is Slump_(economics), which is not the correct sense for this mention. However, in our approach, we only identify five main senses for *slump* and correctly disambiguate this mention to Slump_(sports). One of our observations is that techniques that focus on building their probabilistic models on word co-occurrences on Wikipedia pages tend to favor the senses that have longer descriptions on Wikipedia, although they all normalize based on document length. This seems to be the case in this example as well. In cases when there are too many senses being evaluated by these methods and one of the senses has a relatively longer Wikipedia article, e.g. as is the case for Slump_(economics), then these techniques including TagMe are misled. This problem is avoided in our work where a smaller number of senses are considered.

It is worth noting that we identified the dominant senses from only a corpus of 8M tweets. There are two limitations with this Twitter corpus that could have impeded our work:

- The basic hypothesis of our work is that a tweet should be annotated based on the dominant senses of ambiguous terms on Twitter. This hypothesis implicitly carries that fact that dominant senses of terms can change based on time. Therefore, ambiguous terms within a tweet would need to be annotated with dominant senses detected within the time period when the tweet was posted. However, given the fact that we were interested to compare with the state-of-the-art gold standard, there may have been temporal mis-alignment between the tweets in the gold standard and our Twitter corpus that could have affected the *precision* of our work. The best performance of our work will be achieved when the corpus and the tweets that are being annotated belong to the same time period and hence there is alignment between the dominant senses and the tweets.

- Furthermore, given the limited size of our Twitter corpus, which contained only 8M tweets, it is possible that some of the ambiguous terms in the gold standard were not observed in the Twitter corpus at all, which impacted our performance. As an example, for the tweet *'@yosoyjuanson are you REALLY in tasmania?? go to the MONA MUSEUM!! email me & i'll tell you who to talk to there!!!'*, the detected mention *tasmania* does not exist in our Twitter Corpus, therefore, we were not

able to either identify the mention or detect any of its senses. This impacts the performance of our model in terms of *recall*. However, we believe as the size of our Twitter corpus increases, more accurate results will be obtained.

In summary, our proposed work is an unsupervised method that generates dominant senses from the context of Twitter without relying on all senses from other knowledge bases and yet produces results that are competitive with the state-of-the-art. Based on the observed performance and comparison with the state-of-the-art, we conclude that we can positively respond to our research question and conclude that the consideration of the dominant senses for ambiguous terms on Twitter can positively enhance the semantic annotation of tweets.

4.3 Summary

In this section, we have shown empirical evaluations of our two proposed approaches, separately. In order to evaluate our semantic relatedness method, first, to better understand our tweet corpus, we statistically analyzed our tweet corpus, we obtained the distributions of user's unique words, user's tweets count and co-occurrences between words. Then, we used Spearman's rank correlation and Mean Absolute Error to compare our results with selected state-of-the-art methods and showed that our method has promising results. Additionally, we adopted another well-known evaluation method for semantic relatedness measurement that is employing semantic relatedness method in an application and compared the application's performance. In this work, we created a tweet search engine which uses semantic relatedness method to retrieve results. We incorporated our semantic relatedness method and another state-of-the-art method in a tweet search engine and compared the performance of the search engine in which our method achieves better Mean Average Precision, Reciprocal Rank and Precision at 100. Finally, in order to show that our semantic relatedness method is feasible especially for twitter content, we showed that by using our semantic relatedness method, we can find top related terms for a hashtag which does not have explicit meaning to describe it. We presented a hashtag and its descriptive words to participants who agreed that the descriptive words were very relevant.

We also evaluated our proposed entity linking method. To do so, we first calculated the precision, recall and f1 scores of our method as well as selected state-of-the-art techniques on a public available dataset and showed that our method can obtain comparable results by only considering dominant senses. Since our dominant sense detection method is based on a twitter corpus which is from a different time period compared to the gold

standard dataset, this may cause the tempral-misalignment problem, therefore, we systematically created a benchmark by random sampling tweets from the twitter corpus we used to determine dominant senses and asked volunteers to annotate it. Then we calculated the precision, recall and f1 scores of our method and a selected state-of-the-art technique and showed that if the twitter corpus we use for dominant senses detection is in the same time period of the tweets to be annotated, our method achieves better results which also indicates that our method can capture temporal information on Twitter. Finally, we wanted to show that considering only dominant senses for annotation can be more efficient, therefore, we implemented related techniques which have published their implementation on our local server and performed a paired t-test on the execution time and demonstrated that by using dominant senses, the time can be significantly shortened.

Chapter 5

Conclusion

In this thesis, we have first proposed a novel approach for computing semantic relatedness between two words on Twitter by looking at word co-occurrences on this social network. We have conducted three different types of experiments to assess how well our approach is able to identify the semantics of words within the context of Twitter and measure the semantic relatedness of two words. We have shown that semantics of some words may shift when used on Twitter. Therefore, the state of the art semantic relatedness techniques that focus on encyclopedic knowledge sources are not able to accurately identify the semantics of words in the Twitter context and therefore, would not be ideal for application on this platform. Our proposed approach is able to not only identify the semantics of dictionary words on Twitter but also to capture their semantic shift. In addition, it is able to semantically describe new words on Twitter that do not have formal dictionary semantics such as Internet slang.

Secondly, we have proposed a semantic entity linking method for tweets. Unlike other state-of-the-art techniques that consider all the senses of an ambiguous term from knowledge bases such as Wikipedia, we focus solely on the dominant senses of ambiguous terms mined from Twitter. In order to identify dominant senses, we exploit the *latent relation* hypothesis where by context terms for an ambiguous term are clustered to represent the senses for that term. Once the dominant senses for an ambiguous term are determined, we map each sense onto its corresponding Wikipedia entity. Based on the identified senses and their mapping to Wikipedia entities, we can link tweet mentions to their senses. Using a public available gold standard dataset, we have been able to show that our method has a competitive performance to other baselines including some recently proposed methods in terms of precision and recall even though our Twitter corpus was limited in size and not-directly temporally aligned with the gold standard.

5.1 Future Work

In terms of the semantic relatedness approach, there are two avenues of future work that we would like to explore. First, we are interested in studying whether using a broader context for words on Twitter would impact the quality of the semantic relatedness measure. In the current form, TSSR considers words in the same tweet to have the same context. We would like to expand the context to cover words from the tweets of one user in the same day, or words in a tweet and all of its responses. Second, we are also interested in automatically deriving different senses of a word based on its context on Twitter. In the current form, each word, regardless of how many senses it may have, is represented as a single node in our graph. As future work, we will try to determine the multiple senses of a single word so that semantic relatedness can be measured more accurately.

As for the tweet annotation approach, we are interested in expanding it in three main directions:

- Our work rests on the hypothesis that a limited set of senses for an ambiguous term emerges within a specific time period on Twitter. However, there is no guarantee that this set of senses will remain dominant over time. In other words, as time passes, the set of dominant senses for an ambiguous term might also evolve depending on real-world events and users' interests. Therefore, as our future work, we are interested in exploring the evolution of dominant senses for ambiguous terms. Furthermore, we would also like to study whether it would be possible to find the appropriate length of the time intervals (windows) for which dominant senses will be valid.

- Given the fact that existing techniques such as TagMe and Liu's method consider all Wikipedia senses for an ambiguous term, and our observation that dominant senses can play a positive role in reducing the sense space, we are interested in applying the notion of dominant senses to limit the exploration space of these methods and observe the outcome. TagMe's source code is openly available; therefore, our next step would be to modify TagMe to only consider the dominant senses when performing entity linking on a Tweet as opposed to considering all possible senses from Wikipedia.

- One of the important steps of our work is the identification of the most semantically related terms to an ambiguous term. For this purpose, we build the term dependency graph to find such terms. There has been recent work on word embed-

dings such as *Word2vec* [79] and *GloVe* [89], which use deep neural nets to build vector representations for words. The vector representations preserve very interesting geometric properties for word relatedness and can be used for identifying semantically related terms. We are interested in exploring these works for finding the most related terms for an ambiguous term in our future work.

- It has been shown in [105] that Wikipedia concepts that occur more often in Web search results are also more central to the Wikipedia graph, and are more visited in the Wikipedia web pages. The same thing might happen in Twitter. We are interested in eliminating the least visited Wikipedia concepts and, as a result, the less central Wikipedia entries, and see if focusing on Twitter senses provides any additional advantages.

Bibliography

[1] 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 *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27. Association for Computational Linguistics, 2009.

[2] D. Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 1027–1035, 2007.

[3] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[4] S. Banerjee and T. Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In *Computational linguistics and intelligent text processing*, pages 136–145. Springer, 2002.

[5] E. Biçici. Referential translation machines for quality estimation. Association for Computational Linguistics, 2013.

[6] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[7] 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.

[8] D. Bollegala, Y. Matsuo, and M. Ishizuka. Disambiguating personal names on the web using automatically extracted key phrases. *FRONTIERS IN ARTIFICIAL INTELLIGENCE AND APPLICATIONS*, 141:553, 2006.

[9] D. Bollegala, Y. Matsuo, and M. Ishizuka. Measuring semantic similarity between words using web search engines. *www*, 7:757–766, 2007.

[10] F. Bu, Y. Hao, and X. Zhu. Semantic relationship discovery with wikipedia structure. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1770, 2011.

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

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

[13] A. E. Cano, G. Rizzo, A. Varga, M. Rowe, M. Stankovic, and A.-S. Dadzie. Making sense of microposts:(# microposts2014) named entity extraction & linking challenge. In *CEUR Workshop Proceedings*, volume 1141, pages 54–60, 2014.

[14] K. Chakrabarti, S. Chaudhuri, T. Cheng, and D. Xin. A framework for robust discovery of entity synonyms. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1384–1392. ACM, 2012.

[15] M.-W. Chang, B.-J. Hsu, H. Ma, R. Loynd, and K. Wang. E2e: An end-to-end entity linking system for short and noisy text. *Making Sense of Microposts*, 2014.

[16] H.-H. Chen, M.-S. Lin, and Y.-C. Wei. Novel association measures using web search with double checking. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 1009–1016. Association for Computational Linguistics, 2006.

[17] T. Cheng, H. W. Lauw, and S. Paparizos. Entity synonyms for structured web search. *IEEE transactions on knowledge and data engineering*, 24(10):1862–1875, 2012.

[18] Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: a content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM inter-*

national conference on Information and knowledge management, pages 759–768. ACM, 2010.

[19] R. L. Cilibrasi and P. Vitanyi. The google similarity distance. *Knowledge and Data Engineering, IEEE Transactions on*, 19(3):370–383, 2007.

[20] F. Corcoglioniti, M. Dragoni, M. Rospocher, and A. P. Aprosio. Knowledge extraction for information retrieval. In *International Semantic Web Conference*, pages 317–333. Springer, 2016.

[21] 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*, pages 249–260. International World Wide Web Conferences Steering Committee, 2013.

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

[23] J. Cuzzola and E. Bagheri. Derive: Finding semantic concepts with property-values from natural language text. In *Proceedings of 24th Annual International Conference on Computer Science and Software Engineering*, CASCON '14, pages 331–334, Riverton, NJ, USA, 2014. IBM Corp.

[24] J. Daiber, M. Jakob, C. Hokamp, and P. N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, pages 121–124. ACM, 2013.

[25] L. Derczynski, D. Maynard, G. Rizzo, M. van Erp, G. Gorrell, R. Troncy, J. Petrak, and K. Bontcheva. Analysis of named entity recognition and linking for tweets. *Information Processing & Management*, 51(2):32–49, 2015.

[26] J. Duan and J. Zeng. Computing semantic relatedness based on search result analysis. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 03*, pages 205–209. IEEE Computer Society, 2012.

[27] S. T. Dumais. Latent semantic analysis. *Annual review of information science and technology*, 38(1):188–230, 2004.

[28] C. Fellbaum. *WordNet*. Wiley Online Library, 1998.

[29] Y. Feng, H. Fani, E. Bagheri, and J. Jovanovic. Lexical semantic relatedness for twitter analytics. In *27th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2015, Vietri sul Mare, Italy, November 9-11, 2015*, pages 202–209, 2015.

[30] 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*, pages 1625–1628. ACM, 2010.

[31] F. Ferrara and C. Tasso. Evaluating the results of methods for computing semantic relatedness. In *Computational Linguistics and Intelligent Text Processing*, pages 447–458. Springer, 2013.

[32] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM, 2001.

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

[34] W. A. Gale, K. W. Church, and D. Yarowsky. One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language*, pages 233–237, 1992.

[35] 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. *Proceedings of the VLDB Endowment*, 6(11):1126–1137, 2013.

[36] W. H. Gomaa and A. A. Fahmy. A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13), 2013.

[37] J. Gracia and E. Mena. Web-based measure of semantic relatedness. In *Web Information Systems Engineering-WISE 2008*, pages 136–150. Springer, 2008.

[38] M. Graham, A. Milanowski, and J. Miller. Measuring and promoting inter-rater agreement of teacher and principal performance ratings. *Online Submission*, 2012.

[39] S. Guo, M.-W. Chang, and E. Kiciman. To link or not to link? a study on end-to-end tweet entity linking. In *HLT-NAACL*, pages 1020–1030, 2013.

[40] I. Gurevych. Using the structure of a conceptual network in computing semantic relatedness. In *Natural Language Processing–IJCNLP 2005*, pages 767–778. Springer, 2005.

[41] I. Gurevych. Computing semantic relatedness across parts of speech. Technical report, Technical report, Darmstadt University of Technology, Germany, Department of Computer Science, Telecooperation, 2006.

[42] B. Hachey, W. Radford, J. Nothman, M. Honnibal, and J. R. Curran. Evaluating entity linking with wikipedia. *Artificial intelligence*, 194:130–150, 2013.

[43] L. Han, A. Kashyap, T. Finin, J. Mayfield, and J. Weese. Umbc ebiquity-core: Semantic textual similarity systems. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*, volume 1, pages 44–52, 2013.

[44] 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.

[45] X. Han, L. Sun, and J. Zhao. Collective entity linking in web text: a graph-based method. In *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 765–774, 2011.

[46] X. Han and J. Zhao. Nlpr_kbp in tac 2009 kbp track: a two-stage method to entity linking. In *Proceedings of Test Analysis Conference 2009 (TAC 09)*. Citeseer, 2009.

[47] B. Hecht, S. H. Carton, M. Quaderi, J. Schöning, M. Raubal, D. Gergle, and D. Downey. Explanatory semantic relatedness and explicit spatialization for exploratory search. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 415–424. ACM, 2012.

[48] G. Hirst and D. St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An electronic lexical database*, 305:305–332, 1998.

[49] 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.

[50] 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*, pages 782–792. Association for Computational Linguistics, 2011.

[51] H. Huang, Y. Cao, X. Huang, H. Ji, and C.-Y. Lin. Collective tweet wikification based on semi-supervised graph regularization. In *ACL (1)*, pages 380–390, 2014.

[52] T. Hughes and D. Ramage. Lexical semantic relatedness with random graph walks. In *EMNLP-CoNLL*, pages 581–589, 2007.

[53] I. Hulpuş, 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*, pages 442–457. Springer, 2015.

[54] M. Jarmasz and S. Szpakowicz. Roget's thesaurus: A lexical resource to treasure. *arXiv preprint arXiv:1204.0258*, 2012.

[55] J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*, 1997.

[56] A. Karanastasi and S. Christodoulakis. The ontonl semantic relatedness measure for owl ontologies. In *Digital Information Management, 2007. ICDIM'07. 2nd International Conference on*, volume 1, pages 333–338. IEEE, 2007.

[57] A. Krizhanovsky and F. Lin. Related terms search based on wordnet/wiktionary and its application in ontology matching. *arXiv preprint arXiv:0907.2209*, 2009.

[58] 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*, pages 457–466, 2009.

[59] G. F. Lawler and V. Limic. *Random walk: a modern introduction*, volume 123. Cambridge University Press, 2010.

[60] 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.

[61] J. Lehmann, S. Monahan, L. Nezda, A. Jung, and Y. Shi. Lcc approaches to knowledge base population at tac 2010. In *Proc. TAC 2010 Workshop*, 2010.

[62] C. W. Leong and R. Mihalcea. Measuring the semantic relatedness between words and images. In *Proceedings of the Ninth International Conference on Computational Semantics*, pages 185–194. Association for Computational Linguistics, 2011.

[63] 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.

[64] Y. Li, S. Tan, H. Sun, J. Han, D. Roth, and X. Yan. Entity disambiguation with linkless knowledge bases. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1261–1270. International World Wide Web Conferences Steering Committee, 2016.

[65] D. Lin. An information-theoretic definition of similarity. In *ICML*, volume 98, pages 296–304, 1998.

[66] T. Lin, O. Etzioni, et al. Entity linking at web scale. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 84–88. Association for Computational Linguistics, 2012.

[67] J. Liu, P. Dolan, and E. R. Pedersen. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces*, pages 31–40. ACM, 2010.

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

[69] K. Massoudi, M. Tsagkias, M. De Rijke, and W. Weerkamp. Incorporating query expansion and quality indicators in searching microblog posts. In *Advances in information retrieval*, pages 362–367. Springer, 2011.

[70] Y. Matsuo, J. Mori, M. Hamasaki, T. Nishimura, H. Takeda, K. Hasida, and M. Ishizuka. Polyphonet: an advanced social network extraction system from the web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(4):262–278, 2007.

[71] P. McNamee and H. T. Dang. Overview of the tac 2009 knowledge base population track. In *Text Analysis Conference (TAC)*, volume 17, pages 111–113, 2009.

[72] 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.

[73] C. M. Meyer and I. Gurevych. To exhibit is not to loiter: A multilingual, sense-disambiguated wiktionary for measuring verb similarity. In *COLING*, pages 1763–1780, 2012.

[74] M. Michelson and S. A. Macskassy. Discovering users' topics of interest on twitter: a first look. In *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, pages 73–80. ACM, 2010.

[75] 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*, pages 233–242. ACM, 2007.

[76] R. Mihalcea and D. I. Moldovan. A method for word sense disambiguation of unrestricted text. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 152–158. Association for Computational Linguistics, 1999.

[77] P. Mika. Ontologies are us: A unified model of social networks and semantics. In *The Semantic Web–ISWC 2005*, pages 522–536. Springer, 2005.

[78] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[79] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119, 2013.

[80] T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751, 2013.

[81] N. Milikic, J. Jovanovic, and M. Stankovic. Discovering the dynamics of terms' semantic relatedness through twitter. In *# MSM*, pages 57–68. Citeseer, 2011.

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

[83] D. Milne. Computing semantic relatedness using wikipedia link structure. In *Proceedings of the new zealand computer science research student conference*, pages 1–8, 2007.

[84] J. Mori, M. Ishizuka, and Y. Matsuo. Extracting keyphrases to represent relations in social networks from web. In *IJCAI*, volume 7, pages 2820–2827, 2007.

[85] 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.

[86] N. Nakashole, G. Weikum, and F. Suchanek. Patty: a taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1135–1145. Association for Computational Linguistics, 2012.

[87] S. Patwardhan and T. Pedersen. Using wordnet-based context vectors to estimate the semantic relatedness of concepts. In *Proceedings of the EACL 2006 Workshop Making Sense of Sense-Bringing Computational Linguistics and Psycholinguistics Together*, volume 1501, pages 1–8. Citeseer, 2006.

[88] T. Pedersen. Duluth: Measuring degrees of relational similarity with the gloss vector measure of semantic relatedness. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 497–501. Association for Computational Linguistics, 2012.

[89] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543, 2014.

[90] O. Phelan, K. McCarthy, and B. Smyth. Using twitter to recommend real-time topical news. In *Proceedings of the third ACM conference on Recommender systems*, pages 385–388. ACM, 2009.

[91] G. Pirró. Reword: Semantic relatedness in the web of data. In *AAAI*, 2012.

[92] G. Polčicová and P. Návrat. Semantic similarity in content-based filtering. In *Advances in Databases and Information Systems*, pages 80–85. Springer, 2002.

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

[94] K. Radinsky, E. Agichtein, E. Gabrilovich, and S. Markovitch. A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web*, pages 337–346. ACM, 2011.

[95] 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.

[96] M. E. Renda, M. Bursa, A. Holzinger, and S. Khuri. Information technology in bio-and medical informatics.

[97] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*, 1995.

[98] G. Rizzo, A. E. Cano, B. Pereira, and A. Varga. Making sense of microposts (# microposts2015) named entity recognition & linking challenge. In *5th International Workshop on Making Sense of Microposts (# Microposts 15)*, 2015.

[99] G. Rizzo, M. van Erp, and R. Troncy. Benchmarking the extraction and disambiguation of named entities on the semantic web. In *LREC*, pages 4593–4600, 2014.

[100] M. Röder, R. Usbeck, S. Hellmann, D. Gerber, and A. Both. N3-a collection of datasets for named entity recognition and disambiguation in the nlp interchange format. *9th LREC*, 2014.

[101] H. Rubenstein and J. B. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 1965.

[102] M. Sabou, J. Gracia, S. Angeletou, M. dAquin, and E. Motta. *Evaluating the semantic web: A task-based approach*. Springer, 2007.

[103] M. Sahami and T. D. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th international conference on World Wide Web*, pages 377–386. AcM, 2006.

[104] M. Sanderson. Christopher d. manning, prabhakar raghavan, hinrich schütze, *Introduction to Information Retrieval*, cambridge university press 2008. ISBN-13 978-0-521-86571-5, xxi + 482 pages. *Natural Language Engineering*, 16(1):100–103, 2010.

[105] C. Santamaría, J. Gonzalo, and J. Artiles. Wikipedia as sense inventory to improve diversity in web search results. In *Proceedings of the 48th annual meeting of the association for computational Linguistics*, pages 1357–1366. Association for Computational Linguistics, 2010.

[106] H. Schütze. Automatic word sense discrimination. *Computational linguistics*, 24(1):97–123, 1998.

[107] N. Seco, T. Veale, and J. Hayes. An intrinsic information content metric for semantic similarity in wordnet. In *ECAI*, volume 16, page 1089, 2004.

[108] 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, 2015.

[109] 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.

[110] G. Spanakis, G. Siolas, and A. Stafylopatis. A hybrid web-based measure for computing semantic relatedness between words. In *Tools with Artificial Intelligence, 2009. ICTAI'09. 21st International Conference on*, pages 441–448. IEEE, 2009.

[111] R. Speck and A.-C. N. Ngomo. Ensemble learning for named entity recognition. In *International Semantic Web Conference*, pages 519–534. Springer, 2014.

[112] M. Strube and S. P. Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In *AAAI*, volume 6, pages 1419–1424, 2006.

[113] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.

[114] E. F. Tjong Kim Sang and F. De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the sev-*

enth conference on Natural language learning at HLT-NAACL 2003-Volume 4, pages 142–147. Association for Computational Linguistics, 2003.

[115] T. Tran, N. K. Tran, A. T. Hadgu, and R. Jäschke. Semantic annotation for microblog topics using wikipedia temporal information. 2014.

[116] D. Turdakov and P. Velikhov. Semantic relatedness metric for wikipedia concepts based on link analysis and its application to word sense disambiguation. 2008.

[117] P. Turney. Expressing implicit semantic relations without supervision. 2006.

[118] P. D. Turney. The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, pages 615–655, 2008.

[119] 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.

[120] R. Usbeck, A.-C. N. Ngomo, M. Röder, D. Gerber, S. A. Coelho, S. Auer, and A. Both. Agdistis-graph-based disambiguation of named entities using linked data. In *International Semantic Web Conference*, pages 457–471. Springer, 2014.

[121] M. van Erp, P. Vossen, R. Agerri, A.-L. Minard, M. Speranza, R. Urizar, E. Laparra, I. Aldabe, and G. Rigau. Annotated data, version 2 deliverable d3. 3.2.

[122] B. Vélez, R. Weiss, M. A. Sheldon, and D. K. Gifford. Fast and effective query refinement. In *ACM SIGIR Forum*, volume 31, pages 6–15. ACM, 1997.

[123] J. Waitelonis, C. Exeler, and H. Sack. Linked data enabled generalized vector space model to improve document retrieval. In *Proceedings of NLP & DBpedia 2015 Workshop in Conjunction with 14th International Semantic Web Conference (ISWC 2015). CEUR Workshop Proceedings*, 2015.

[124] S. Wan and R. A. Angryk. Measuring semantic similarity using wordnet-based context vectors. In *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pages 908–913. IEEE, 2007.

[125] C. Welty, J. W. Murdock, A. Kalyanpur, and J. Fan. A comparison of hard filters and soft evidence for answer typing in watson. In *International Semantic Web Conference*, pages 243–256. Springer, 2012.

[126] I. Witten and D. Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia*

and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA, pages 25–30, 2008.

[127] H. Wu, M. R. Min, and B. Bai. Deep semantic embedding. In *SMIR@ SIGIR*, pages 46–52, 2014.

[128] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994.

[129] M. Yahya, K. Berberich, S. Elbassuoni, M. Ramanath, V. Tresp, and G. Weikum. Natural language questions for the web of data. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 379–390. Association for Computational Linguistics, 2012.

[130] I. Yamada, H. Takeda, and Y. Takefuji. An end-to-end entity linking approach for tweets. In *CEUR-WS*, 2015.

[131] D. Yang and D. M. Powers. *Verb similarity on the taxonomy of WordNet*. Masaryk University, 2006.

[132] E. Yeh, D. Ramage, C. D. Manning, E. Agirre, and A. Soroa. Wikiwalk: random walks on wikipedia for semantic relatedness. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, pages 41–49. Association for Computational Linguistics, 2009.

[133] T. Zesch. *Study of semantic relatedness of words using collaboratively constructed semantic resources*. PhD thesis, TU Darmstadt, 2010.

[134] T. Zesch and I. Gurevych. Automatically creating datasets for measures of semantic relatedness. In *Proceedings of the Workshop on Linguistic Distances*, pages 16–24. Association for Computational Linguistics, 2006.

[135] T. Zesch, C. Müller, and I. Gurevych. Using wiktionary for computing semantic relatedness. In *AAAI*, volume 8, pages 861–866, 2008.

[136] W. Zhang, J. Su, C. L. Tan, and W. T. Wang. Entity linking leveraging: automatically generated annotation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1290–1298. Association for Computational Linguistics, 2010.

[137] Q. Zhao, S. C. Hoi, T.-Y. Liu, S. S. Bhowmick, M. R. Lyu, and W.-Y. Ma. Time-dependent semantic similarity measure of queries using historical click-through data. In *Proceedings of the 15th international conference on World Wide Web*, pages 543–552. ACM, 2006.

[138] W. Zhou, H. Wang, J. Chao, W. Zhang, and Y. Yu. Loddo: Using linked open data description overlap to measure semantic relatedness between named entities. In *The Semantic Web*, pages 268–283. Springer, 2011.

[139] X. Zou, C. Sun, Y. Sun, B. Liu, and L. Lin. Linking entities in tweets to wikipedia knowledge base. In *Natural Language Processing and Chinese Computing*, pages 368–378. Springer, 2014.