SEMANTIC ANALYSIS USING WIKIPEDIA GRAPH STRUCTURE

by

Armin Sajadi

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

at

Dalhousie University
Halifax, Nova Scotia
February 2018

Table of Contents

List of Tables

List of Figures

Abstract

Wikipedia is becoming an important knowledge source in various domain specific applications based on concept representation. While lexical resources like WordNet cover generic English well, they are weak in their coverage of domain-specific terms and named entities, which is one of the strengths of Wikipedia. Furthermore, semantic relatedness methods that rely on the hierarchical structure of a lexical resource are not directly applicable to the Wikipedia link structure, which is not hierarchical and whose links do not capture well defined semantic relationships like hyponymy. We introduce a vector space representation of concepts using Wikipedia graph structure to calculate *semantic relatedness*. The proposed method starts from the neighbourhood graph of a concept as the primary form and transfers this graph into a vector space to obtain the final representation. The proposed method achieves state-of-the-art results on various relatedness datasets. We evaluate Wikipedia in a domain-specific semantic relatedness task and are able to demonstrate that Wikipedia-based methods can be competitive with state of the art ontology-based methods and distributional methods in the biomedical domain. The comparison includes a wide range of structure and corpus-based methods, such as our proposed word2vec-based embeddings: a hybrid distributional/knowledge-based word2vec and $node\text{-}embedding$, a word2vec application on graph structure. Our representations have also been reported to achieve the highest results in a query expansion task.

We also use a standard *coherence model* to show that the proposed relatedness method performs successfully in Word Sense Disambiguation (WSD). We then suggest a different formulation for coherence to demonstrate that, in a short enough sentence, there is one *key entity* that can help disambiguate every other entity. Using this finding, we provide a vector space based method that can outperform the standard coherence model in a significantly shorter computation time. We use our findings in WSD to create a complete *wikifier*, a supervised approach based on *learning to rank* that combines our new *coherence measure* with other sources of information, such as textual context. The final product is an open source project that is available through direct API or web service.

List of Abbreviations and Symbols Used

FP_i	False Positives when resolving the ith document.
$I(v)$	Incoming neighbours of v.
L	Laplacian.
$N_G^{-/+}[v]$	Closed in-/out-neighbourhood graph of v.
$O(v)$	Outgoing neighbours of v.
S	Given sentence to be disambiguated.
TP_i	True positives when resolving the ith document.
ϵ_i	possible *key entity*.
erfc	Complementary Error Function.
$\hat{\mathcal{E}}$	List of entities, a possible disambiguation.
$\hat{\mathcal{R}}(\cdot)$	Context vector w.r.t mention m.
$\hat{\pi}^M$	Macro-averaged precision.
$\hat{\pi}^M$	Macro Average Precision.
$\hat{\pi}^\mu$	Micro-averaged precision.
$\hat{\pi}^\mu$	Micro-Averaged Precision.
$\hat{e}_i$	Possible entity associated to the ith mention..
$\mathcal{B}_i$	List of resolved entities, if ϵ_i is the correct *key entity*.
$\mathcal{C}$	Set of all candidates.
$\mathcal{C}_i$	List of candidates for the ith mention.
$\mathcal{E}$	List of correct entities.
$\mathcal{E}^*$	List of entities, found by the disambiguation.
$\mathcal{M}$	List of mentions.
$\mathcal{R}(\cdot)$	Vector representation of an entity/mention.
$\vec{1}$	All-ones vector.
b_i^j	Entity associated with the jth mention, if ϵ_i is the correct *key entity*.
c_i^j	j-th candidate for mention i.

$conf(\cdot)$	Confidence value for the best candidate for mention m_i, when disambiguating using the context similarity algorithm.
$depth(v)$	Depth of the node v.
e^*	*Key entity*.
e_i	Correct entity for the ith mention.
k_i	Number of candidates for mention i.
k_i^j	j-th best candidate for the ith mention, w.r.t to the context of the ith mention..
lcs	Least Common Subsumer.
m_i	i-th mention in the sentence.
$n(t)$	Number of occurrences of t.
$path(u, v)$	length of the path between u and v.
$r(\cdot, \cdot)$	Semantic relatedness measure.
z_ρ	Fisher's z-transformation of ρ.
BOW	Bag Of Word.
CUI	Concept Unique Identifier.
HITS	Hyperlink-Induced Topic Search.
IC	Information Content.
IIC	Intrinsic Information Content.
ILP	Integer Linear Programming.
IP	Integer Programming.
IR	Information Retrieval.
KB	Knowledge Base.
LCH	Leacock-Chodorow.

LKR	Lexical Knowledge Resource.
LP	Linear Programming.
LSA	Latent Semantic Analysis.
NEL	Named Entity Linking.
NER	Named Entity Recognition.
NGD	Normalized Google Distance.
NGED	Normalized Graph Edit Distance.
NLM	National Library of Medicine.
NLP	Natural Language Processing.
POS	Part Of Speech.
PPR	Personal Pagerank.
RG	Rubenstein and Goodenough.
SVD	Singular Value Decomposition.
SVM	Support Vector Machine.
TFIDF	Term FrequencyInverse Document Frequency.
UMLS	Unified Medical Language System.
VSM	Vector Sense Model.
WLM	Wikipedia Link Measure.
WN	WordNet.
WP	Wu & Palmer.
WSD	Word Sense Disambiguation.

Acknowledgements

I would first like to express my sincere gratitude to my advisors, Dr. Evangelos E. Milios and Dr. Vlado Keselj. I am indebted to them for their continued support and patience throughout my PhD and for giving me this opportunity to pursue my interest in NLP and guidance. In addition, I would like to thank Dr. Jeannette Janssen, who has taught me a lot, and whose comments greatly improved my work.

I would like to extend my appreciation to Dr. Norbert Zeh for his generous and detailed comments on the whole dissertation, and my external examiner, Dr. Virendra C. Bhavsar, for accepting to attend my defence and providing valuable feedbacks. A special mention goes to Chris Maxwell, for his quick solutions to the numerous technical problems I faced in the course of my thesis.

I also want to acknowledge Ryan Amaral for his contribution to this work. Ryan implemented the Wikificantion module as part of his USRA project, and kindfully reviewed my last chapter. I am also very thankful to Paige Lana E Black whom kindly proofread parts of the manuscript.

I am very grateful to Dr. Robin Gras, to whom I am greatly indebted. He will be always an example for me throughout my career.

I would like to thank my fellow lab members, Dr. Raheleh Makki, Dr. Magdalena Jankowska and Dr. Axel Soto for their feedback, support and friendship. Without them, something would have been missing from this long journey.

I am also grateful to my friend, Saman Vaisipour. Saman's advice, persuasion and sometimes bitter logic played a significant role in every decision I made during my education. Without him, I would not be standing where I am today.

The most painful part of the journey was being without my family (Mamanoo Adanoo Payam) next to me, and the sweetest part was having Shali beside me. The confidence to continue is solely attributable to their unconditional love and moral and emotional support and I cannot imagine finishing my PhD without them.

Chapter 1

Introduction

"Concepts[1] are the constituents of thought" [124]. We address the problem of finding a *concept representation* that can be used in basic Natural Language Processing (NLP) tasks, such as *semantic relatedness*, *word sense disambiguation* (WSD) or document annotation.

The traditional way of representing a concept is based on *set theory*, attributed to Aristotle. In this view, a concept can be represented by a set with its elements being its *attributes*. For example, a bird can be defined by a set of properties such as {*has-wings, can-fly, lays-eggs*}. It is obvious that finding this set is not always easy. For example, an *Ostrich* cannot fly and still is a bird, while a *butterfly* has all these properties and yet is not a bird. The history of philosophy is full of efforts to define concepts in terms of their essential properties, referred to as *essentialists*. It took a long time until *Ludwig Wittgenstein* developed his *Family resemblance* theory in which he states that no such set exists; items (here, different birds) have only overlapping similarities. This theory gained a lot of attention in psychology and many experiments proved its correctness [174]. An equivalent theory following family resemblance is *cue validity* theory (a.k.a. conditional probability of features) which replaces the binary relationship between properties and concepts with conditional probabilities [173]. Many other forms of representations have been suggested and the most popular one is the *vector space* model, sometimes referred to in cognitive domain as *spatial representation* [69]. In cognitive linguistics, the basic dimensions are chosen to be basic attributes, referred to as *quality dimensions* [59] and the space is called *conceptual space*. In more pragmatic views, the constraints on dimensions are relaxed and they can be anything, from the *term-set* [179] to *hidden* (or *imaginary*, *latent*) concepts in LSA [101].

The elements of this vector can be inferred either by incorporating expert knowledge expressed in the form of ontologies, or distributional analysis of an unstructured corpus.

[1] In computational linguistics, the border between what is called a concept in cognitive science (such as *bird*) or an instance or object of such concept (such as *Sparrow*) is fuzzy and they are all referred to as a concept.

Ontologies are human-curated and reliable; however, they have several limitations:

1. Creating and maintaining quality domain ontologies is labour-intensive; such ontologies are maintained only in some critical domains, such as the biomedical domain.

2. Consequently, many ontologies are far from complete, and an attempt has been made to maintain only very crucial relations, mostly taxonomic relations (IS-A).

Distributional approaches are a way to overcome these limitations. Typically, to find the dimensions, the term-set of the collection is used and for the projection, co-occurrence in a context is usually the core idea. But the main limitation of distributional methods is their need to have access to sufficiently large corpora to be competitive [1]. Although corpus creation is less costly than an ontology, compiling a suitable corpus for a specific domain is still not a trivial task.

Wikipedia, on the other hand, is getting more popular and evaluating it in different tasks shows its acceptance by the NLP and IR communities. Comparing Wikipedia with the other two resources reveals some of its strengths:

1. It is cheap, as a result of crowdsourcing, and yet studies suggest it is very reliable [203, 40, 169].

2. Features are human-curated, and not being restricted to a specific relation makes them very rich.

3. Provides *Literary Warrant* [77], a term used to denote the explanation provided to the user in justification of a relatedness.

4. Wikipedia covers a wide range of domains. This is the main feature of Wikipedia that makes it a suitable resource for domain-specific analysis.

As a result, the first problem we are addressing in this manuscript is the suitability of Wikipedia for domain-specific semantic analysis.

The second motivation is to use the Wikipedia structure to represent a concept, formulated as follows: we are given the Wikipedia graph and the objective is to find a suitable *embedding*, that is, a vector representation for each concept. We make two important assumptions in our approach:

1. The *graph* structure of Wikipedia suffices to represent the concepts.
2. The neighbourhood of a concept contains all the information needed to represent the concept.

Regarding the first assumption, we have to note that using the Wikipedia graph can be seen as a kind of feature selection. In other words, we conjecture that ignoring non-concept words does not affect the quality. Also the graph structure is richer in some aspects; many low-frequency concepts do not co-occur with other words, but they do play a role in the graph structure, due to their links to high-frequency words and also categories.

The second assumption can be justified from a computational point of view. The Wikipedia graph has a very small diameter and expanding a node more than a single level can lead to a blow up on the number of nodes.

The intuition behind our proposed method is to use the nodes in the neighbourhood of a concept and rank them using the structure of the graph. For example; *September 2008* and *Clozapine* are both connected to *Schizophrenia*, where the former is just a date when new statistics about behavioural disorders were published, and the latter is a drug to treat *Schizophrenia*. The whole process can be illustrated with an example, for a diverse entity such as *Noam Chomsky*.

To asses the suitability of Wikipedia and the quality of our representations at the same time, we evaluate our vectors in a very basic task in concept-based text processing: *semantic relatedness* in the biomedical domain. Given a pair of concepts, semantic relatedness is defined to be a real value representing any possible taxonomic or non-taxonomic relationship between them. Having a vector representation, calculating the semantic relatedness can be done using any well-known metric.Biomedical domain has two important advantages over other domains: first, there exist several well-known ontologies, all human-curated and actively used and maintained by domain experts. Second, calculating semantic relatedness is a well explored problem in this domain and a rich literature exists on different ontology-based and corpus-based methods. We compare our semantic relatedness methods with various ontology-based, corpus-based and hybrid methods. The evaluations are usually done on ground truth datasets, consisting of several pairs of concepts along with a real value, expressing the degree of relatedness between each pair.

The quality of these datasets can be an issue in the evaluations; they are usually limited in size and vulnerable to subjective errors. Also semantic relatedness is usually going to be

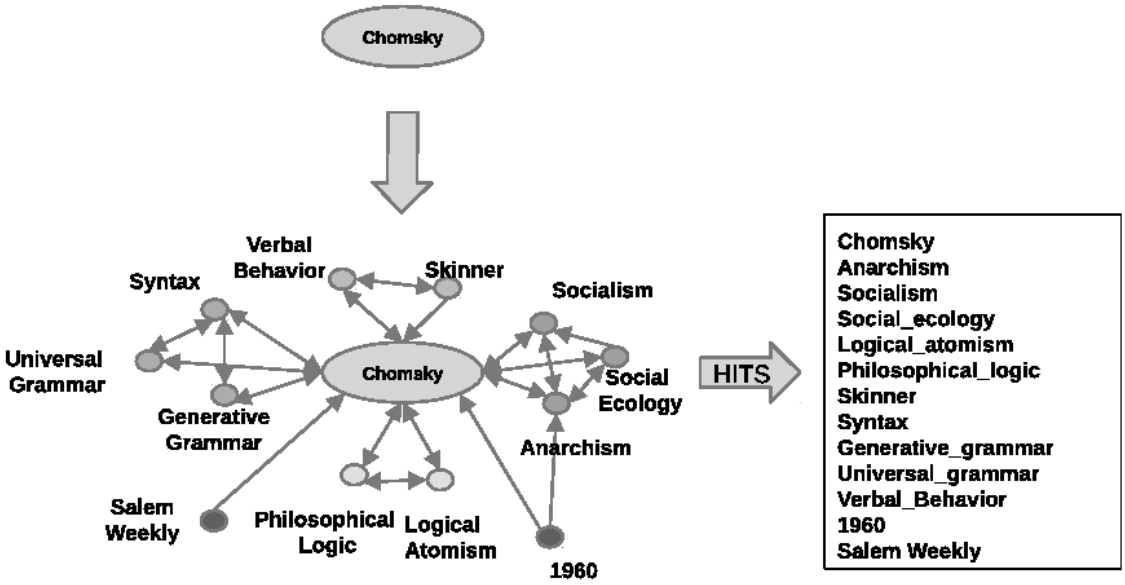


Figure 1.1: Illustration of the steps of finding a vector representation for the concept of *Noam Chomsky*

used as an inner module of a more practical NLP tool. This motivated us to perform extrinsic evaluations by using the concept representations in other NLP tasks. We first report a successful incorporation of our embeddings in a "Microblog Filtering System" [118, 119]. The task is approached using a classic information retrieval approach, and several semantic relatedness methods are tried in the *query expansion phase*. The results demonstrate that our representation achieves the best performance compared with the other approaches.

However, most of our evaluation is focused on the *word sense disambiguation* (WSD) task. Words in a sentence can have multiple senses, classic examples are "*bass*" (a *type of fish* or a *musical term*) or "*bank*" (*sloping land* or a *financial institute*). WSD is the problem of identifying the correct senses for all ambiguous words in a given sentence. By limiting ourself to only the concept representations and ignoring other useful information that can help disambiguation, we tend to compare our semantic relatedness method with other methods regarding this task. We start from a standard model, called *coherence model*, which among possible senses for a word, picks the ones that maximizes the sum of pairwise similarities of the words. Being an NP-complete problem, we use Integer Programming

(IP) to solve the optimization. Our semantic relatedness method outperforms popular state-of-the-art methods on multiple datasets.

We later observed that in a short sentence, there is usually one entity that once disambiguated, can help to disambiguate every other entity . We refer to this entity as the *key entity*. Using this notion, we redefine the *coherence measure* to be the sum of similarities of all entities in the sentence to the *key entity*. In our first attempt, we try to solve the new optimization directly, which leads to a quadratic solution, a big improvement over the original NP-complete problem. We refer to this new approach as *key entity based word sense disambiguation*. We demonstrate that not only is the new optimization much easier to solve, it results in an improvement of the final results.

In a second attempt, we try to use the vector space model to guess this *key entity* directly. This is achieved by simple vector space operations to model the *context vector* and *confidence* of a disambiguation. The goal of this VSM-based key entity recognition is to show how having vector representations can be beneficial, as opposed to a black-box semantic relatedness module. Compared to the aforementioned entity disambiguation methods (coherence and quadratic *key entity* recognition), our VSM-based *key entity* method shows to be as successful, and in most cases, more accurate, and more than one hundred times faster.

With the previous modules discussed, we almost have all the necessary components to implement a Wikifier. Wikifiers usually consist of two main steps: *mention detection* (extracting the entities) and *link* them to Wikipedia. We evaluated several methods for the first step and our final approach is a combination of two models: a Finite State Transducer (FST) for detection and a supervised learning model, trained by thousands of examples, for pruning.

For the entity linking phase, we combine our disambiguation method with the textual information and some other statistics that we deliberately ignored in the previous experiments. Regarding textual context, we evaluated multiple methods, including a simple TFIDF [122] and state-of-the-art *word2vec* [134]. We do this by taking multiple features from the context and our coherence measure, and feed it to a "*rank learner*". Learning to rank is a supervised model that, given enough instances of ranked lists with their features, learns to rank new instances. Learning to rank performs better than a regular classifier in our experiments. The final product, *Wikisim*, as a web service, along with all the data and

the source code, is released as an open source project [2].

1.1 Contributions

The outcome of this research is a collection of methods that can facilitate concept-based document processing on several levels. The contributions of this research are:

1. A comparative analysis between Wikipedia, biomedical ontologies and biomedical corpora, in the semantic relatedness task. The results demonstrate that Wikipedia outperforms the domain-specific resources, and therefore may be an adequate resource, especially for domains lacking proper ontologies or corpora.

2. Introducing a hybrid relatedness model of ontology and *neural embedding* (word2vec) for the biomedical domain. This model achieves state-of-the-art results on some of the datasets and serves as a strong baseline in our comparisons.

3. Proposing a new vector representation for concepts using the graph structure of Wikipedia. The new method is based on neighbourhood embedding of the concepts. Our representation is evaluated in three NLP tasks: Semantic Relatedness, Query Expansion and Word Sense Disambiguation. We compare a wide range of graph embeddings, as well as distributional and more recent neural embeddings.

4. Proposing a new *coherence measure*, based on the idea of *key entity*. This approach conjectures that there exists a key entity in a short sentence that can assist in disambiguating other entities. Our model can be optimized in less computation time and outperforms the previous measure on various datasets.

5. Implementing a Wikifier, a complex system with several modules for entity recognition and disambiguation and linking to Wikipedia. It combines our concept based method with the textual information using a supervised *learning to rank* algorithm.

1.2 Outline

The comparison between Wikipedia and biomedical ontologies is done in Chapter 2 (published as [191, 181]). We also compare Wikipedia with our hybrid semantic relatedness,

[2] `https://github.com/asajadi/wikisim`

which is a word2vec trained embedding on a normalized corpus with the aid of biomedical entity linkers. This method is the first application and adaptation of word2vec for the biomedical domain, and is used as a baseline in Pakhomov et al. [149], and McInnes and Pedersen [128]. In this chapter, we also report our first attempt to calculate semantic relatedness using Wikipedia. The introduced method has two limitations: (1) the representation is not a vector, therefore we need to use a rather complicated list metric. (2) it does not provide a powerful concept representation, the final result is a combination of two calculated relatedness scores and works like a black box (referred to as *HITS-sim*).

We overcome the two limitations in Chapter 3, which is an extension of an already published work [180]. When different embeddings are thoroughly reviewed, we can see that one specific method (rvsPagerank) can provide high-quality vectors. We extend our comparisons and add more baselines in this chapter, some evaluated for the first time on Wikipedia, such as graph based distance, neural node embedding (an application of word2vec on graphs), and *normalized Google distance*. We also report the evaluation of incorporating our embeddings in *query expansion* for microblog filtering in this chapter.

In Chapter 4, we evaluate our concept vectors in WSD. The content of this chapter is also an extension of [180]. We first try *coherence* method, a general and *similarity-agnostic* method. Later we introduce our two versions of key entity-based disambiguation and demonstrate that this method can outperform the traditional coherence method, in a significantly shorter computation time.

In Chapter 5, we introduce a complete Wikifier. This system applies our concept-based embedding and disambiguation to a realistic project, combining it with other sources of information such as the textual data of Wikipedia, which we have ignored so far. The combination is done using state-of-the-art *learning to rank* algorithms.

We provide an open-source web service to facilitate incorporating the system into NLP projects. In Appendix A, we discuss the API of the system, as well as a high level introduction to the technologies used in the implementation. The initial version of our implementation won *Verifiability, Reproducibility, and Working Description Award (1st place)* in the 15th *International Conference on Computational Linguistics and Intelligent Text Processing* [63].

Chapter 2

Domain-Specific Semantic Relatedness from Wikipedia Structure

2.1 Introduction

Semantic relatedness is a relationship between a pair of concepts. This relation can be the well known taxonomic relation (i.e., *is-a*) or any non- taxonomic relation such as antonymy, meronymy (*is-a-part-of*) or domain specific relations, such as *is-treated-by* and *is-caused-by* in the biomedical domain. We address the problem of quantifying the relatedness into a real value to be used in applications such as query expansion, word sense disambiguation, and information retrieval. A detailed review of the applications is given in [25].

Most concept-based information retrieval systems in the biomedical domain rely on ontologies to calculate relatedness. Ontologies are labour-intensive to create and do not exist for most domains. Where ontologies are unavailable, an alternative is using distributional (a.k.a. corpus-based) methods. However, distributional methods can only be competitive if they have access to sufficiently large domain-specific corpora [1, 61]. Building such corpora for many domains is not trivial.

This project assesses the suitability of Wikipedia as a potential knowledge resource for semantic relatedness computation and compares it to three classes of methods: (1) methods using domain-specific human-authored biomedical ontologies (2) state-of-the-art distributional methods and (3) a hybrid of ontology based and distributional methods that we build by using the recent developments in deep learning for distributional representation. The third method outperforms corpus-based methods previously reported in the literature. We focus on biomedical domain because of the availability of high-quality ontologies (MeSH, SNOMED-CT, etc.), a rich literature for extracting semantic relatedness [156, 61], successful distributional methods and corpora [156, 99, 194], and reliable datasets [156, 147, 148].

To calculate relatedness, we present a novel method that takes advantage of the prepared concept *graph* structure in Wikipedia. By focusing only on the Wikipedia graph, we implicitly assume that the only relevant phrases in the text of the concept pages are those linking to other concepts (a.k.a. anchor texts). We also make an explicit assumption, that

the only relevant features for representing a concept c are its neighbours in the Wikipedia graph, or in other words, those mentioned in the page associated with c and/or those which mention c in their pages. These pages are human-curated and the relevance is always explained in the text, so any attempt to use concepts not mentioned in the text disregards the explanatory structure of Wikipedia and lacks the notion of *Literary Warrant* [77]. Based on these assumptions, the intuition behind the proposed algorithm is to use the concepts in the neighbourhood of a concept and rank them using the structure of the graph. For example, while *September 2008* and *Clozapine* are both connected to *Schizophrenia*, the former is just a date when some new statistics about behavioural disorders were published and should be ranked lower than the latter that is a drug to treat *Schizophrenia*.

The contributions of this research are:

1. Comparing Wikipedia against ontologies and distributional methods in estimating relatedness, thereby demonstrating that Wikipedia may be a suitable knowledge resource for calculating relatedness in domains lacking such high quality resources

2. Adapting and evaluating a group of structure-based graph similarity methods of various degrees of sophistication on Wikipedia, motivated by the non-hierarchical structure of Wikipedia,

3. Evaluating the recent dense word embedding called word2vec [134, 133, 135] in the biomedical domain for the first time. We also propose a hybrid method that combines word2vec and biomedical knowledge source. This method achieves the highest results reported in the literature for some of the datasets, and has been cited and used as a state-of-the-art baseline by the leading researchers in the field of biomedical text analysis (i.e, Pakhomov et al. [149], and Bridget McInnes and Ted Pedersen [128]).

4. Proposing a new similarity method based on the idea of ranking the neighbours and evaluating its performance.

All evaluations are performed on datasets containing pairs of biomedical terms and a gold standard semantic similarity value for each pair. The results are compared with the results of the ontology-based methods using well known biomedical ontologies as their knowledge source, as well as distributional methods on well known corpora.

2.2 Related Work

We summarize the literature of semantic relatedness in three sections: (i) the original problem in general domain (ii) the biomedical domain and (iii) methods that use Wikipedia as their knowledge source.

2.2.1 Relatedness in General Domain

Approaches for computing semantic relatedness are traditionally categorized as *distributional* (a.k.a. corpus-based), *Lexical Knowledge Resource* (LKR) based (LKR can refer to dictionary, taxonomy or ontology) or hybrid if they use both at the same time. This categorization often obscures the fact that LKR can have content (other than structure) and plays the role of a corpus as well. For example, *Extended Gloss Overlap* [15] is known as LKR based, while it is using both the structure and the definitions (known as glosses) of WordNet. In this case, WordNet plays the role of both the corpus and the LKR at the same time. This specific method needs a corpus to operate [153], while having access to an LKR is optional [156]. There are other methods with the opposite characteristics for which a corpus can be useful but not necessary [83]. We use the label *structure-based* to describe methods using only the structure of a knowledge base, typically via graph-representation. A list of well known semantic relatedness methods along with the resources they use is given in Table 2.1.

From another perspective, methods can be either unsupervised or supervised, regarding whether they have access to human-labelled data. Some examples of the approaches used by unsupervised methods are: path information [205, 104], Information Content (IC) [170, 90, 186], Latent Semantic Analysis (LSA) [102], n-gram [112, 87], context similarity methods [15, 153, 1], Normalized Google Distance method (NGD) [39] and Personalized Pagerank (PPR) algorithm [83, 4]. Regarding methods based on WordNet only, the state-of-the-art ones are the Context Vector method [156], which uses the glosses, and the Personalized Pagerank (PPR) method [83, 1], which is based on structure. Corpus-based methods can produce competitive results using large datasets and computational resources [1]. The best reported results are obtained using hybrid methods on a web corpus and WordNet [1]. The most successful supervised method is using Support Vector Machine (SVM) [4].

2.2.2 Relatedness in the Biomedical Domain

The majority of studies in relatedness in the biomedical field concentrate on ontology-based methods, as such methods benefit from the availability of high quality manually curated ontologies. Two well known ontologies are Systematized Nomenclature of Medicine-Clinical Terms (SNOMED-CT) and Medical Subject Headings (MeSH). These resources can be accessed directly or through a framework called Unified Medical Language System (UMLS), in which these ontologies and several other terminologies are integrated. Most of the methods applied on these ontologies are successful WordNet-based methods [156, 61]. However, there are a few methods developed specifically for the biomedical domain, (e.g., [145]).

As in the general domain, distributional methods can obtain competitive results in the biomedical domain, again depending on the quality of the corpus. The distributional approaches presented in [156, 99, 194] show promising results on small test datasets (although the last two approaches are in fact hybrid). Some studies suggest that on larger test datasets, ontology-based methods outperform distributional methods by a wide margin [61].

2.2.3 Relatedness from Wikipedia

Wikipedia as a resource for semantic relatedness has been evaluated on well known domain-independent datasets. Different methods are either adaptations of ontology-based (WikiRelate [161]), distributional (Explicit Semantic Analysis (ESA) [56]), structure-based (Wikipedia Link Measure (WLM) [202], *Visiting probability* (VP) [208]) or hybrid (WikiWalk [209]) methods. WikiRelate adapts an ontology-based method to the category structure of Wikipedia. However, it is significantly outperformed by similar ontology based methods. ESA happens to be the best single-resource based method [1]. WLM follows ESA in evaluations [202, 209] and has been used in recent projects [183]. Similar to the general case, Wikipedia-based methods can be categorized as either distributional or structure-based, depending on whether they use the text of Wikipedia or its structure.

One common problem in the evaluation of the mentioned systems is ignoring the fact that Wikipedia is an encyclopedia, not a dictionary and therefore, general words are not covered as well as domain-specific terms. The only domain specific evaluation [206] is focused on text similarity rather than concrete word similarity. On the contrary, this evaluation method is neither on a standard dataset nor against an ontology.

General Domain	LKR	Corpus	Approach
Wu and Palmer 1994 (WUP) [205]	✓	✗	Path Based
Resnik 1995 [170]	✓	✓(WN Glosses)	IC
Jiang and Conrath 1997 (jcn) [90]	✓	✓(WN Glosses)	IC
Landauer 1998 [102]	✗	✓	LSA
Leacock and Chodorow 1998 (lch) [104]	✓	✗	Path Based
Lin 1998 (Lin) [111]	✓(WN Glosses)	✓	C
Lin 2003 [112]	✗	✓	N-gram
Banerjee and Pedersen 2003 [15]	✓	✓(WN Glosses)	Extended Gloss Overlap
Seco et al. 2004 [186]	✓	✗	Intrinsic IC
Patwardhan and Pedersen 2006 [153]	✓/✗	✓(WN Glosses)	Context Vector
Bollegala et al. 2007 [22]	✗	✓	SVM (Supervised)
Cilibrasi and Vitanyi 2007 (NGD) [39]	✗	✓	NGD
Hughes and Ramage 2007 [83]	✓	✓/✗	PPR
Agirre et al. 2009 [1]	✗	✓	Context Window
Iosif and Potamianos 2010 [86]	✗	✓	Search Engine Results
Agirre et al. 2009 [1]	✓	✗	PPR
Islam et al. 2012 [87]	✗	✓(Google n-gram)	N-gram
Mikolov et al. 2013 [134]	✗	✓(Wikipedia Text)	SkipGram
Biomedical Domain)	LKR	Corpus	Approach
Rada et al. 1989 [166]	✓	✗	Path Based
Caviedes, and Cimino 2004 (Cdist) [30]	✓	✗	Path Based
Nguyen and Al-Mubaid 2006 (NAM) [145]	✓	✗	Path Based
Sánchez and Batet 2011 [182]	✓	✗	Intrinsic IC Based (Path+IC)

Table 2.1: A summary of the literature of relatedness methods. The first group was proposed for general domain but most of them have also been evaluated on the biomedical domain. The second group are those originally proposed for the biomedical domain.

Our similarity method is also related to a method [117] proposed for a different task, namely for calculation of similarity between publications based on the citation graph. This method uses the authority scores assigned by HITS [98]. Aside from the difference between domains (citation analysis and concept relatedness), there are three main other differences: First, we use the neighbourhood graph only. Second, we use all scores returned by HITS, not only authority scores. Third, we use a different distance calculation (a comparison is presented in Section 2.6.9).

2.3 Similarity Measures

In this section, we briefly introduce the similarity measures that we use in our study. The two larger categories are *structure-based* and *distributional*. However, within structure-based method we make a distinction between taxonomic methods, i.e, methods that assume a tree structure for the knowledge source, and *graph-based* methods, which use the whole graph (all relations) in the ontology. Within Distributional methods, we introduce three methods: *Context Vector* method [156], a well-known method in the biomedical domain and ESA [56], a method based on Wikipedia and Word2vec [134].

2.3.1 Taxonomic Structure-Based Measures

There are various ways of using the path length between two concepts in an ontology, including simply the *path* ([156]), Wu & Palmer method [205] and Leacock & Chodorow method [104]. This set of measures is widely used when the knowledge source has a taxonomic structure. However, they have been tried on Wikipedia using its category structure [161], with a limited success.

Path [166]

Below is the simplest way to measure the similarity between two given concepts a and b. It is defined as the inverse of the length of the path between them.

$$r_{path}(a,b) = \frac{1}{len(a,b)}. \tag{2.1}$$

LCH [104]

LCH is essentially a scaling of r_{path} by the depth of the ontology. Let d be the depth of the ontology, then it is defined as

$$r_{LCH}(a,b) = 1 - \frac{log(len(a,b))}{log(2 \times d)}. \tag{2.2}$$

Wu & Palmer [205]

There are different versions of this measure. The most widely used one is as follows: Let $lcs(a,b)$ be the *least common subsumer* of a and b. r_{wp} is defined to be the depth of this node, scaled by their individual depth

$$r_{wp}(a,b) = \frac{depth(lcs(a,b))}{depth(a) + depth(b)}. \tag{2.3}$$

However, in one of the other baselines, Garla et al. [61] use a different version of WP that gives a perfect similarity of 1 for the similarity of a concept with itself.

$$r_{wp}(a,b) = \frac{2 \times depth(lcs(a,b))}{len(a,b) - 1 + 2 \times depth(lcs(a,b))}. \tag{2.4}$$

Using Information Content

One modification to the path based methods is to include the effect of the *information content* (IC) of a node. Information content can be calculated from an external source, or in our case, directly from the ontology [182, 61]:

$$IC_{intrinsic}(c) = -\log\left(\frac{\frac{|leaves(c)|+1}{|subsumers(c)|}}{\max(leaves(c)) + 1}\right), \tag{2.5}$$

where $leaves(c)$ is the set of all leaves descending from c and $subsumers(c)$ is the set of the ancestors of c, including c. Information content measures the information a node carries.

One way to directly use information content is to use the Lin measure [111]

$$r_{lin}(a,b) = \frac{2 \times IC(lcs(a,b))}{IC(a) + IC(b)}. \tag{2.6}$$

To redefine LCH using this new concept, we need to re define both path and depth (d). To do so, $len(a,b)$ is replaced by the information content-based distance [90]

$$len_{jc}(a,b) = IC(a) + IC(b) - 2 \times IC(lcs(a,b)), \tag{2.7}$$

where $lcs(a, b)$ is the *least common subsumer* of a and b. Then d is also replaced by ic_{max}, the maximum information content across all concepts. Finally, the intrinsic information content based LCH (IIC-LCH) is defined as

$$IIC\text{-}LCH(a, b) = 1 - \frac{log\,(len_{jc}(a, b) + 1)}{log\,(2 \times ic_{max} + 1)}. \tag{2.8}$$

The new information-based path length can be used to redefine $r_{path_{jc}}$:

$$r_{path_{jc}}(a, b) = \frac{1}{len_{jc}(a, b) + 1}. \tag{2.9}$$

2.3.2 Graph-Based Methods

Methods Based on Random Walks

The earliest studies investigating the possibility of using Random Walk for calculating semantic relatedness are [83] and [16]. The idea is to perform two random walks for given nodes u and v, each of the walks *personalized* (or *customized*) for one of the target nodes, resulting in two different distributions. Treating a discrete distribution as a vector, the similarity would be the distance between them; cosine similarity is often used. customized random walk for a node v is a random walk that restarts from v after several iterations. The result is a limiting probability of finding the walker on each node of the graph, assuming that the starting point was v. Another way to interpret the probability distribution is to see it as the importance (or score) of each node of the graph, from the viewpoint of v.

Several studies, mainly by Hughes [83] and Eneko Agirre [1, 4, 2], have applied this method and obtained state-of-the-art results for the general and biomedical domains.

Wikipedia Linked-Based Measure (WLM)

WLM [202] is the most widely used similarity measure on Wikipedia. It has been incorporated in many text mining tasks such as *Named Entity Recognition* [168, 75, 19, 105] and *Link Prediction* [200]. It is the application of another successful similarity metric, *Normalized Google Distance* (NGD) [39] on Wikipedia, and is one of the strongest baselines. NGD originally counts occurrences of terms, denoted by $n(t)$ for the term t. WLM uses the same equation but counts only those occurrences of t that are anchored (linked) to the Wikipedia page associated with the concept, i.e., $I(a)$ if a is the Wikipedia page. For any

pair of concepts a and b, Wikipedia Link-Based Measure of the two terms is defined as Eq. 2.10.

$$WLM(a,b) = \frac{\log(\max(|I(a)|,|I(b)|)) - \log(|I(a) \bigcap I(b)|)}{\log(|V|) - \log(\min(|I(a)|,|I(b)|))}, \tag{2.10}$$

2.3.3 Distributional Methods

Explicit Semantic Analysis (ESA)

Wikipedia contains a large amount of text and can be used as a corpus. Explicit Semantic Analysis (ESA) [56] is a very simple approach that, using the concept article text, extracts the term-document matrix and uses the term vectors in the document space as the representation.

Context Vector Method [156]

Context Vector is one of the first and most widely used hybrid relatedness methods in the biomedical domain. It calculates a representation for each concept in three steps:

1. A co-occurrence matrix is built for each word (using a fixed size window) from a large corpus of text (Mayo Clinic Corpus of Clinical Notes with 1,000,000 clinical notes).

2. Every SNOMED-CT concept c that occurs more than a threshold in the corpus is looked up in a thesaurus, namely Mayo Clinic Thesaurus, and its description is extracted. Mayo Clinic Thesaurus is a thesaurus maintained since 1909, containing 5,167,428 unique phrases.

3. Every word in the description is looked up from the co-occurrence matrix, and they are aggregated to form the context vector for concept c.

It is obvious that this method uses both a corpus and two knowledge sources, making it a hybrid method. At the end, the semantic relatedness is calculated using the cosine similarity.

2.3.4 Word2vec

Word2vec [134, 133] is the state of art among distributional methods. It is a simple (one-layer) neural network based method that learns a vector for each word in such a way that it is close to words co-occurring with it, while at the same time far from those that are not. The algorithm has two different versions, *Continuous Bag of Words* (CBOW) and *Skip-gram*. In CBOW, the goal is to predict a word given its context. In this approach the position of the words are discarded (hence called BOW), and the average of the context words are assumed to be the context representation. In SkipGram, on the other hand, contexts of words are being predicted using the current word (the process is illustrated in Fig. 2.1). Skip-gram gives consistently better results in the original experiments [134] and in more recent studies [109].

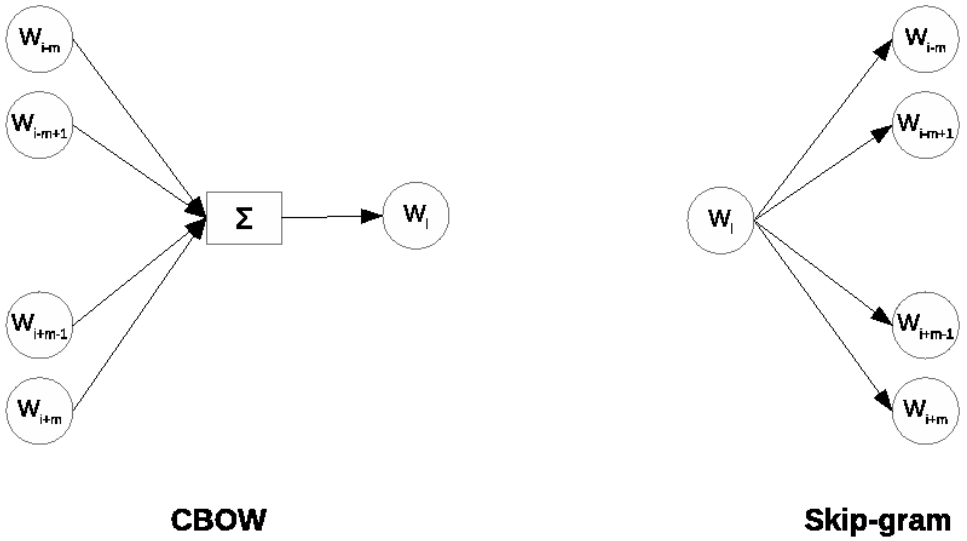


Figure 2.1: Word2vec word embedding CBOW: Predicting a given word w_i given the context words $w_{i-m}, w_{i-m+1} \cdots, w_{i+m-1}, w_{i+m}$, Skip-gram: predicting context words $w_{i-m}, w_{i-m+1} \cdots, w_{i+m-1}, w_{i+m}$ given w_i [134]

2.4 Wikipedia Graph

A Wikipedia *page* is associated with each *concept*, so a directed graph can be obtained with nodes representing concepts and edges representing out links from one page to another.

Definition 2.1 *The Basic Wikipedia graph is a digraph $G_b(V_b, E_b)$ where V_b is the set of Wikipedia concepts and $(u, v) \in E_b$ iff there is a link from the page associated with u pointing to the page associated with v.*

There is a specific type of edge, called *redirect*. Redirecting denotes synonymy (for example *UK* is redirected to *United Kingdom*). We derive another graph, called *Wikipedia graph* by defining the concept of *Synonym Ring* of a node, i.e, the set of nodes synonymous to it. The idea is to find a way to group synonym nodes to form a *meta* node and, using it, merge the edges between nodes to become edges between meta nodes:

Definition 2.2 *E_r is defined to be the set of redirections, redirection denotes synonymy:*

$$(u, v) \in E_r \implies (u, v \in E_b) \wedge u \textit{ is a synonym of } v \tag{2.11}$$

Definition 2.3 *The Epsilon closure of a node is the set of the nodes accessible from it by travelling along redirect links:*

$$\varepsilon(v) = \{v\} \cup \{u | \left((v, u) \in E_r\right) \vee \left(\exists u_i \in \varepsilon\left(v\right) \wedge \left((u_i, u) \in E_r\right)\right)\} \tag{2.12}$$

Definition 2.4 *The synonym Ring of a node v is the set of nodes synonymous to v*

$$sr(v) = \{u | \left(u \in \varepsilon\left(v\right)\right) \vee \left(\exists u_i \in sr\left(v\right) \wedge u_i \in \varepsilon\left(u\right)\right)\} \tag{2.13}$$

In this thesis, by referring to a node associated with a concept, we always mean the synonym ring of the node (Fig. 2.2). Finally the Wikipedia graph can be defined.

Definition 2.5 *A Wikipedia graph $G(V, E)$ is the graph on the synonym rings of the nodes of the basic graph defines as follows:*

$$\begin{aligned} V &= \{sr(v) | v \in V_b\} \\ E &= \{(sr(u), sr(v) | (u, v) \in E_b - E_r\} \end{aligned} \tag{2.14}$$

Definition 2.6 *For any digraph $G = (V, E)$ and any node v, We use $I(v)$ to denote the set of in-neighbours of node v and $O(v)$ to denote the set of its out-neighbours. For each node v, we define three sub graphs: (i) $N_G[v]$, the closed neighbourhood graph of v, is the subgraph of G induced by v and all vertices adjacent to v. (ii) $N_G^-[v]$, the closed in-neighbourhood graph of v, is the subgraph induced by v and $I(V)$. (iii) $N_G^+[v]$, the closed out-neighbourhood graph of v, subgraph induced by v and $O(V)$.*

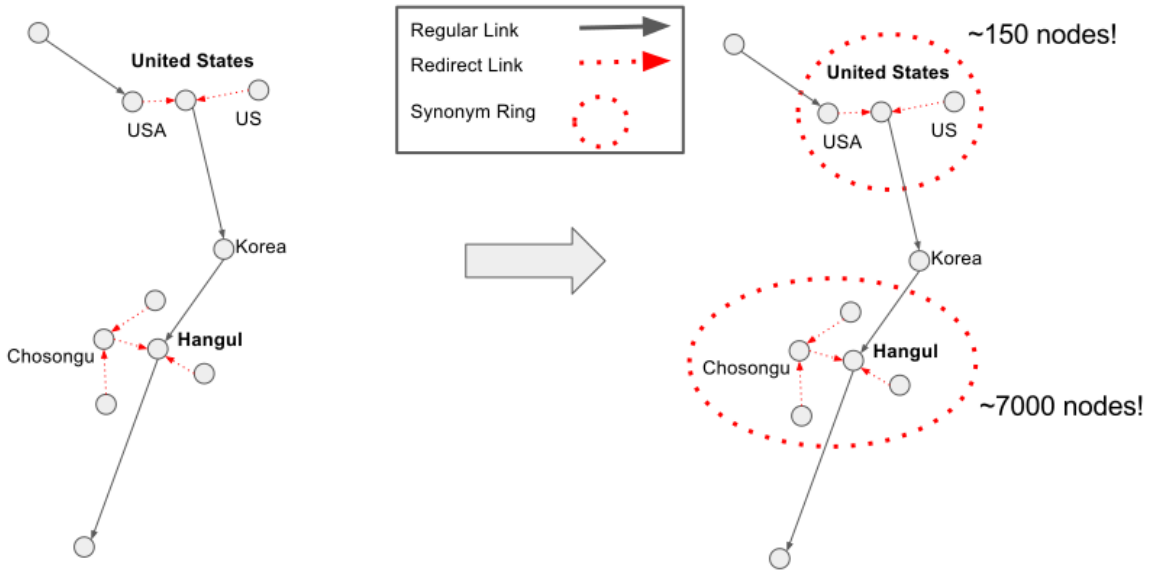


Figure 2.2: Nodes redirecting to each other form a synonym ring

2.5 Methodology

To confirm our conjectures at the beginning of this chapter, we take the following steps:

1. Report the results obtained using state-of-the-art ontology-based and distributional methods on various biomedical datasets.

2. Train our hybrid model on a domain-specific corpus to build a strong "non-Wikipedia" distributional model. This model outperforms previous distributional models in the literature.

3. Apply different node similarity metrics on Wikipedia and evaluate them on the ground truth dataset.

4. Evaluate our proposed graph-based methods with the ground truth datasets.

5. Compare the results of 2 to 1 to demonstrate the success of our distributional method.

6. Compare the results from 4 to 1–3 to demonstrate both the richness of Wikipedia as a knowledge source for the biomedical domain and the success of our proposed graph-based method.

2.5.1 Distributional (and Hybrid) Word2vec for the Biomedical Domain

As explained in Section 2.3.4, word2vec is the most widely used distributional method in the literature. However, no thorough evaluation of it exists so far for the biomedical domain. We use word2vec on a biomedical corpus as a strong alternative to both ontologies and Wikipedia. However, using a pure word2vec method is not very successful, mainly due to the wide variation of biomedical domain terms: each term has many synonyms. So we realized that taking an extra step to normalize different forms of a term will affect the results dramatically. We do this by performing an *entity linking* to UMLS to replace all different surface forms with their universal identifier, known as Concept Unique Identifier (CUI). This method is used as a baseline in Pakhomov et al. [149], and McInnes and Pedersen [128]. We summarize the procedure in the following steps:

1. Corpus preparation: We use a collection of 348,566 references from medical journals over a five-year period (1987–1991), called OHSUMED dataset [76]. This is the same dataset used by *tensor encoding* and other distributional methods (such as [99]).

2. *Entity Normalization*: We use MetaMap [12] to map the phrases to UMLS concepts. MetaMap is a tool, widely used in biomedical research, that detects mentions and disambiguates them. It uses several features and approaches in the process, such as *part of speech tagging*, *syntactic (shallow) parsing*, *mention detection* and *context analysis*. The result is replacing medical terms with their CUIs.

3. Learning the embeddings: We train *word2vec* on this corpus. Our experiments show that this is far superior to using only *word2vec*, or to using the combination of automatic phrase detection and *word2vec*.

2.5.2 Bibliometrics

The Wikipedia graph is not hierarchical, so well known taxonomy-based methods cannot be directly applied. To compute the relatedness between concepts, we start from simple and well known graph-based methods. A straightforward approach to compare two graphs is to calculate their overlap. In our case, both graphs are *vertex-induced subgraphs* of one graph, that is the Wikipedia graph, and hence, overlap measure is simply the vertex overlap. Using bibliographic similarity terminology, when given two concepts a and b, we can count the

portion of common incoming neighbours ($co\text{-}citation$ [190], Eq. 2.15), common outgoing neighbours ($coupling$ [94], Eq. 2.16) and a combination of both through a weighted average of $co\text{-}citation$ and $coupling$ ($amsler$ [11], Eq. 2.17). These are three simple yet powerful methods for the bibliographical domain, as well as hypertext mining [43].

$$co\text{-}citation(a,b) = \frac{|I(a) \cap I(b)|}{|I(a) \cup I(b)|} \tag{2.15}$$

$$coupling(a,b) = \frac{|O(a) \cap O(b)|}{|O(a) \cup O(b)|} \tag{2.16}$$

$$amsler(a,b) = \frac{|(I(a) \cup O(a)) \cap (I(b) \cup O(b))|}{|(I(a) \cup O(a)) \cup (I(b) \cup O(b))|} \tag{2.17}$$

SimRank

The methods discussed so far focus on only incoming or outgoing links and ignore the structure of the graph. If we want to go a step further and consider the relationships among the neighbours, one possibility is using a well known algorithm called SimRank. SimRank [89] is a structural similarity method extending bibliometrics. It can be considered as a generalized version of $co\text{-}citation$ that takes into account the similarities among the citing nodes as well. It can be interpreted as the probability of two surfers meeting in the same node if they start walking backward from a and b [89]. It is a recursive equation (Eq. 2.18) and runs until it converges.

$$\begin{aligned} s_0(a,b) &= 1 \text{ if } a = b, \text{ else } 0 \\ s_{k+1}(a,b) &= \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s_k(I_i(a), I_j(b)) \end{aligned} \tag{2.18}$$

where $I_i(v)$, for $1 \leq i \leq |I(v)|$ denotes individual incoming neighbours of node v and C is a decay factor between 0 and 1.

If the recursion is applied to outgoing links, it is called $rvs\text{-}SimRank$ [211] (Eq. 2.19),

and if applied in both directions, it is called $P\text{-}Rank$ [211] (Eq. 2.20).

$$\begin{aligned} r_0(a,b) &= p_0(a,b) = 1 \text{ if } a = b, \text{ else } 0 \\ r_{k+1}(a,b) &= \frac{C}{|O(a)||O(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} r_k(O_i(a), O_j(b)) \end{aligned} \tag{2.19}$$

$$\begin{aligned} p_{k+1}(a,b) &= \lambda \times \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} p_k(I_i(a), I_j(b)) \\ +(1-\lambda) &\times \frac{C}{|O(a)||O(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} p_k(O_i(a), O_j(b)) \\ \lambda &\in [0,1] \end{aligned} \tag{2.20}$$

where $I_i(v)$, for $1 \leq i \leq |I(v)|$ and $O_i(v)$, for $1 \leq i \leq |O(v)|$ denote in-neighbours and out-neighbours of node v respectively.

SimRank should be applied over the entire Wikipedia graph resulting in all pairwise similarities. Due to scalability limitations and the large size of Wikipedia, we compute $s(a,b)$, $r(a,b)$ and $p(a,b)$ using only the joint-neighbourhood graph of the concepts u and v. Similar to the Definition 2.6, we define in-joint-neighbourhood, $N_G^-[u,v]$, as the subgraph induced by $\{u,v\} \cup I(u) \cup I(v)$, out-joint-neighbourhood, $N_G^+[u,v]$, as the subgraph induced by $\{u,v\} \cup O(u) \cup O(v)$ and joint-neighbourhood, $N_G[u,v]$, as the subgraph induced by $\{u,v\} \cup I(u) \cup I(v) \cup O(u) \cup O(v)$.

Another issue is that nodes with a higher number of neighbours result in a higher similarity. To compensate for this effect, motivated by [89], we propose a final tuning. We need to slightly change the modification in [89] to keep the metric symmetric.

$$s_P(a,b) = s(a,b) \cdot I(a)^P \cdot I(b)^P \tag{2.21}$$

$$r_P(a,b) = r(a,b) \cdot O(a)^P \cdot O(b)^P \tag{2.22}$$

$$p_P(a,b) = p(a,b) \cdot I(a)^P \cdot I(b)^P \cdot O(a)^P \cdot O(b)^P \tag{2.23}$$

$$P \in [0,1]$$

2.5.3 Our Proposed Method: HITS Based Similarity

In this section, we propose our similarity method, which can be considered as another form of extension to basic bibliometric methods. The intuition is: *similar nodes with similar*

rankings in the neighbourhood of the two concepts means high relatedness. The problem with the basic graph overlap calculation is that most nodes have a very high number of neighbours and not all of them have the same importance. Our idea is to rank the neighbours of a node based on the role they play in its neighbourhood. We use Hyperlink-Induced Topic Search (HITS) [98] to do so. HITS is a well known concept in information retrieval. It was originally developed to rank a set of search results, but we use it in a similarity calculation method, referred to as *HITS-Based method* in this project.

Algorithm 1 HITS Based Similarity Computation

1: **function** $HITS\text{-}sim_{st}(a,b,st)$

Input: : a,b, two concepts; $st \in \{\text{HUB}, \text{AUTHORITY}\}$, score type

Output: : Similarity between a and b

2: $N[a] \leftarrow$ Extract a neighbourhood graph for a

3: $N[b] \leftarrow$ Extract a neighbourhood graph for b

4: $L[a] \leftarrow HITS(N[a], st)$ ▷ $L(a)$ will contain neighbours of a sorted by HITS

5: $L[b] \leftarrow HITS(N[b], st)$ ▷ $L(b)$ will contain neighbours of b sorted by HITS

6: $L'[a] \leftarrow append(L[a], reverse(L[b] \setminus L[a]))$

7: $L'[b] \leftarrow append(L[b], reverse(L[a] \setminus L[b]))$

8: **return** $1 - PartialTopKKendall\text{-}Distance(L'[a], L'[b])$

9: **end function**

10: **function** $HITS(N,st)$

Input: : N, An adjacency matrix representing a graph; st, score type

Output: : An ordered list of vertices

11: $Sc \leftarrow$ Using HITS calculation, get the required score (HUB or AUTHORITY) based on st for each node in N

12: $L \leftarrow$ sort vertices of N based on Sc in descending order

13: **return** L

14: **end function**

To compute similarity between two concepts using this idea, we propose Algorithm 1. In steps 2 and 3, neighbourhood graphs can be any of the forms introduced in Section 2.4. In steps 4 and 5, we use the HITS algorithm to obtain a representative list of vertices to use as the basis of relatedness between the two concepts. HITS gives every node two scores: *hub score* and *authority score* through a recursion on the graph. So if it is run over a

graph consisting of pages related to a concept (*focused graph* in HITS terminology), the final product of the algorithm is two ranked lists: authoritative pages and those which are good hubs to the authoritative pages. For the similarity measure, we can use either the hub list $HITS\text{-}sim_{hub}(\cdot,\cdot)$ = $HITS\text{-}sim_{st}(\cdot,\cdot,\text{HUB})$ or the authority list: $HITS\text{-}sim_{aut}(\cdot,\cdot)$ = $HITS\text{-}sim_{st}(\cdot,\cdot,\text{AUTHORITY})$. HITS assigns two initial scores to each node p, *authority score*, $x^{\langle p\rangle}$, and *hub score*, $y^{\langle p\rangle}$, and uses the mutual reinforcement relation between the two scores:

1. The x score of nodes pointed to by nodes with higher y should be higher.

2. The y score of nodes pointing to nodes with higher x should be higher.

Assuming that E is the set of the edges, HITS initializes every score with 1 and performs the following iterations for each node p:

$$x^{\langle p\rangle} \leftarrow \sum_{q:(q,p)\in E} y^{\langle q\rangle} \tag{2.24}$$

$$y^{\langle p\rangle} \leftarrow \sum_{q:(p,q)\in E} x^{\langle q\rangle} \tag{2.25}$$

By normalizing these scores after each step, assuming M to be the adjacency matrix, it is provable that these equations converge and the final value of X, the vector of all x scores, will be the principal eigenvector of M^TM, and the final value of Y, the vector of all y scores, will be the principal eigenvector of MM^T [98].

Extended HITS [187] is another approach that uses the same idea of mutual reinforcement to compute node similarity in a graph. Aside from the two *hub* and *authority* lists, it extracts a third scored list of nodes that can be considered as *intermediating* between *hubs* and *authorities*. We can treat the scores assigned by Extended-HITS the same way we do with HITS in Algorithm 1. We refer to this variation by $EHITS\text{-}sim$. Using either of these scores, we end up representing each concept by a list.

Having two ordered lists after step 5, we are facing a classic ordered list comparison, which can be done by *Kendall's tau* Distance [45]. Kendall's tau works on two lists with the same elements and increases the distance for each pair of elements with different orders in the lists. In steps 6–7, we append the concepts missing in one list and present in the other one, to the list that is missing them. Our motivation in reversing the order is to penalize the similarity for any pair that one or both of them are missing in either of the lists.

Kendall's tau distance calculates the number of pairwise disagreements between the two lists. If σ_1 and σ_2 are two lists, with the same elements (in different orders) and length n, it is defined as:

$$K(\sigma_1, \sigma_2) = \frac{2}{n(n-1)} \sum_{\{i,j\} \in \mathcal{P}} \bar{K}_{i,j}(\sigma_1, \sigma_2) \tag{2.26}$$

where

- $\mathcal{P}$ is the set of unordered pairs of distinct elements of the lists.
- $K_{i,j}(\sigma_1, \sigma_2)$ is 0 if i and j are in the same order in both of the lists; otherwise it is 1.

Hub and Authority capture two different aspects of similarity, so our final score (and our proposed method referred to by $HITS\text{-}sim$) will be a weighted average of both scores to combine them in one similarity score. To avoid parameter tuning, we always use a simple average with $\lambda = 0.5$.

$$\begin{aligned} HITS\text{-}sim(a,b) &= \lambda \times HITS\text{-}sim_{hub}(a,b) \\ &+ (1-\lambda) \times HITS\text{-}sim_{aut}(a,b) \\ &\lambda \in [0,1] \end{aligned} \tag{2.27}$$

2.6 Evaluation

2.6.1 Methods and Parameter Set-ups

For $amsler$, $P\text{-}Rank$ and $HITS\text{-}sim$, we set $\lambda = 0.5$ to make it a simple average. Also for $SimRank$ and its variations, the parameters are taken from the original experiments [89] ($C = 0.8$ and $P = 0.5$). We do not report all variants of SimRank and HITS-based methods; We obtained the best results with the following settings: $SimRank$, $rvs\text{-}SimRank$ and $P\text{-}Rank$ on $N_G^-[\cdot]$, $N_G^+[\cdot]$ and $N_G[\cdot]$ respectively and $HITS\text{-}sim_{aut}$, $HITS\text{-}sim_{hub}$ and $EHITS\text{-}sim$ on $N_G^-[\cdot,\cdot]$, $N_G^+[\cdot,\cdot]$ and $N_G[\cdot,\cdot]$ respectively. All experiments are based on the 20120403 dump version of Wikipedia.

2.6.2 Example

An excerpt from the vector representation of two similar pairs of concepts is shown in Table 2.6.2. We did this experiment using authority score distribution over the in-neighbourhood

graph. In all four examples, the most related concept to the given concept is itself. Note that for the second pair, *Zoloft* and *Prozac* are brand names for *Sertraline* and *Fluoxetine*. We only have shown the first 10 ranked neighbours, but it is still obvious that our ranking method has been able to bring the more important concepts to the top of the list. The distribution of the score over all the incoming neighbours is shown in Figure 2.3. The explanatory nature of Wikipedia make it very easy to explain the relation between concepts. For example with *Prozac* and *Sertraline*, we can find the associated sentence: "*in 2010, over 24.4 million prescriptions for generic formulations of Fluoxetine were filled in the United States, making it the third most prescribed antidepressant after sertraline and citalopram*".

King	**Rook**	**Zoloft**	**Prozac**
King	Rook	Sertraline	Fluoxetine
Chess	Chess	Phenelzine	Sertraline
Glossary of chess	Queen	Tranylcypromine	Venlafaxine
Algebraic notation	Bishop	Nefazodone	Phenelzine
Pawn	Glossary of chess	Aripiprazole	Tranylcypromine
Rook	Pawn	Amoxapine	Nefazodone
Queen	Algebraic notation	Clorgiline	Fluvoxamine
Bishop	Knight	Iproniazid	Duloxetine
Knight	King	Buspirone	Amitriptyline
327	**2118**	**1245**	**1112**

Table 2.2: Ranking the list of neighbours using the proposed method for two similar pairs (*King*, *Rook*) and (*Zoloft*,*Prozac*). Only the top 10 of ranked neighbours for each concept are shown. The last row shows the number of neighbours.

2.6.3 Semantic Relatedness Comparison Metrics and Significance

The standard relatedness datasets are sets of paired concepts with a human-assigned score which is considered to be the ground truth for their relatedness. The more scores reported by the automatic system correlate with the ground truth, the better the system is. The preferred method to measure this is Spearman's rank correlation, denoted by ρ (a.k.a. Spearman's rho). Having two variables X and Y, Spearman's rank correlation is defined to be the Pearson correlation between the ranks. If we define R^X to be the rank vector of X , i.e, R^X_i be the rank of X_i in X, and similarly, R^Y be the rank vector of Y, Spearman's rho can

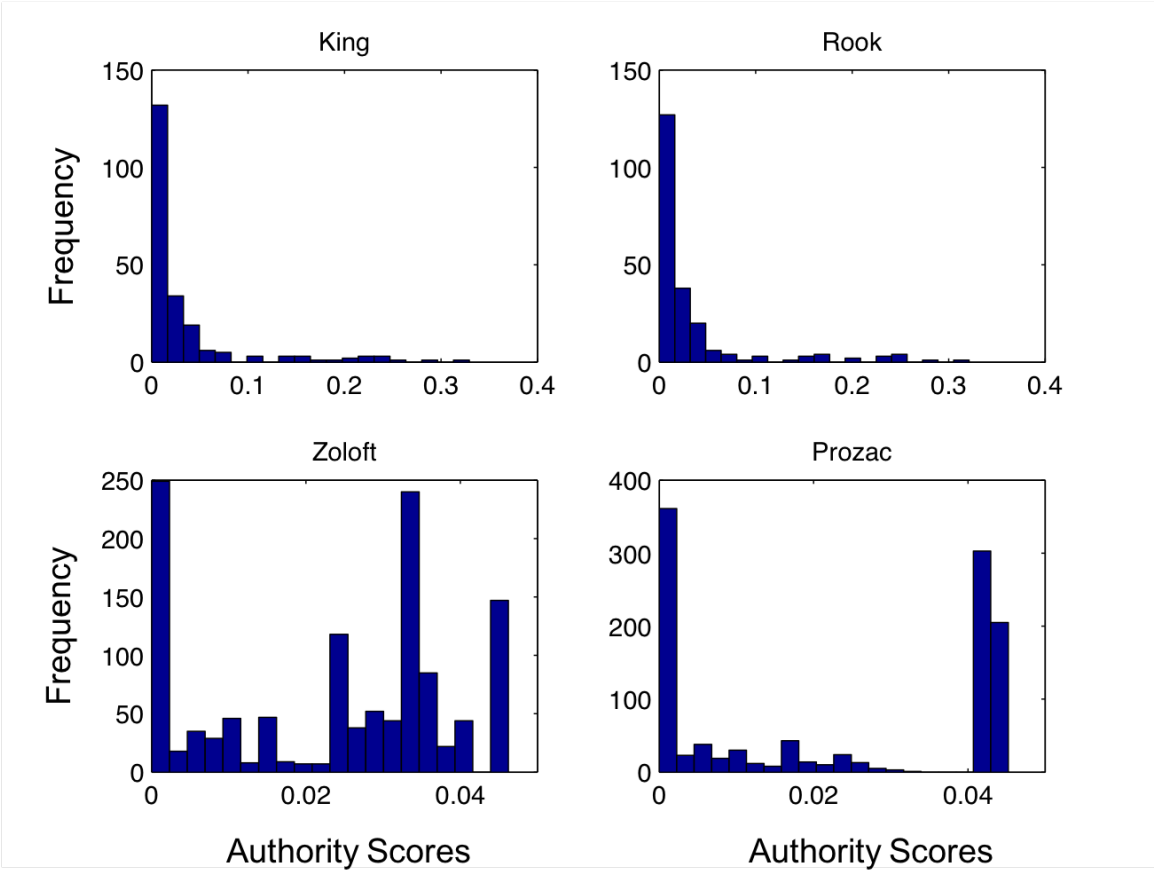


Figure 2.3: Distribution of authority scores for four concept examples

be calculated by Eq 14.6.1 from [162]:

$$\rho = \frac{\sum_i (R_i^X - \bar{R^X})(R_i^Y - \bar{R^Y})}{\sqrt{\sum_i (R_i^X - \bar{R^X})^2}\sqrt{\sum_i (R_i^Y - \bar{R^Y})^2}} \tag{2.28}$$

Few of the published approaches in this field report statistical significance of their results. We use a one-tailed test on the Fisher's $z\text{-}score$ to calculate the significance of correlations [3] when comparing our results to published results of other methods. To calculate significance we converted ρ to z_ρ using Fisher's transform (Eq. 2.29) that is shown to be normally distributed [48].

$$\begin{aligned} z_\rho &= \frac{1}{2}\ln\frac{1+\rho}{1-\rho}, \\ \sigma(z_\rho) &= \sqrt{\frac{1.06}{n-3}} \end{aligned} \tag{2.29}$$

Having z_ρ, one can use one-tailed test on the $z\text{-}score$ to calculate the significance of a correlation by Eq. 14.5.9 [162]:

$$\operatorname{erfc}\left(\frac{z_\rho}{\sqrt{2}\sigma(z_\rho)}\right) \tag{2.30}$$

where erfc is the *complementary error function*:

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}}\int_x^\infty e^{-t^2} \tag{2.31}$$

To calculate the difference between two correlations ρ_1 and ρ_2, we transform them following Eq. 2.29 to obtain $(z_{\rho_1}, \sigma(z_{\rho_1}))$ and $(z_{\rho_2}, \sigma(z_{\rho_2}))$ and use the fact that their difference is a normal distribution with a standard deviation $\sqrt{\sigma^2(z_{\rho_1}) + \sigma^2(z_{\rho_2})}$:

$$\operatorname{erfc}\left(\frac{|z_{\rho_1} - z_{\rho_2}|}{\sqrt{2}\sqrt{\sigma^2(z_{\rho_1}) + \sigma^2(z_{\rho_2})}}\right) \tag{2.32}$$

For hybrid *word2vec* and Wikipedia-based methods, we use a more accurate method for calculating significance, known as Zou's method for *dependent overlapping correlations* [215]. The details of the method is beyond the scope our research, but in summary let's suppose we have three variables, X as the gold standard and Y and Z as the two methods to be compared. In other words, we want to calculate the significance interval for $\rho_{XY} - \rho_{XZ}$. Zou's method requires to have ρ_{YZ} (the correlation between Y and Z) and hence, one needs the actual scores between all pairs for both X and Y to apply this method. We usually do not have access to this information if we report correlations from other studies.

2.6.4 Datasets and Baseline Methods

For general domain evaluation and comparison against WordNet, we use the following standard datasets:

1. Miller and Charles (MC), 30 pairs [137]: This is a subset of another dataset (Rubenstein and Goodenough (RG) list of 65 pairs) that is re-scored by Miller and Charles. The dataset consists of general word-pairs such as (*food*, *fruit*).

2. WordSimilarity-353 collection [50]: Similar to MC, but with more pairs (353 pairs). To distinguish *relatedness* and *similarity* evaluations, Agirre et al. [1] proposed a split of the dataset into two overlapping subsets, referred to by *WordSim353-rel* and *WordSim353-sim* respectively.

Within the biomedical domain, there exist higher quality and reliable datasets of bigger sizes; the increased size of the datasets leads to more significant results:

1. Pedersen benchmark [156]: A set of 29 concepts and the most reliable dataset that biomedical comparisons are usually based on. A set of 120 pairs of concepts was initially rated by 13 indexing experts and pairs with a high agreement were selected to a second round of rating. This time, the pairs were scored by three physicians and nine medical indexing experts (referred to as Coders in the tables).

2. Mayo benchmark [148]: a set of 101 concept pairs ranked by 13 Mayo Medical Index experts. Pakhomov et al. [148] proposed a general framework to compile and evaluate semantic relatedness benchmarks; Mayo dataset is the result of that study.

3. UMN benchmark [147]: A group of medical residents rated a set of 724 pairs of different semantic types (disorders, symptoms and drugs). Each pair was given two different ratings for *similarity* and *relatedness*. The result of several statistical reliability tests was two different overlapping sets of 587 and 566 concepts pairs, focusing on relatedness (referred to by UMN Rel) and similarity (referred to by UMN Sim) respectively.

2.6.5 Knowledge Sources

We compare Wikipedia with the ontologies available in the general UMLS framework, designed by the United States National Library of Medicine (NLM). UMLS provides an environment to integrate several biomedical terminologies, means to translate between them, and links to external knowledge sources [21]. UMLS provides related tools as well, such as MetaMap. The concepts in UMLS can have many different relations with each other in different ontologies. For this reason we need to differentiate between two important links: taxonomic (*is-a*) and non-taxonomic. In the provided statistics in Table 2.3, "*taxonomy*" and "*All*" refer to this classification. In all of the ontology-based methods only taxonomic relations are used while PPR based methods use all relations.

All concepts are identified by their Concept Unique Identifier (CUI) in UMLS; we mapped them to Wikipedia pages manually. We refer to Mesh and SNOMED-CT (through UMLS) by *sct-umls* and *mesh-umls* in the tables. Also by *umls* we refer to MESH, SNOMED-CT and 60 other lexicons, all integrated into UMLS 2.3.

		Taxonomy		All	
Name	Description	Concepts	Relations	Concepts	Relations
sct-umls	UMLS SNOMED CT	284,213	431,393	319,824	1,272,567
msh-umls	UMLS MeSH	315,081	426,139	321,306	1,266,235
umls	All UMLS	1,861,805	2,580,066	2,046,351	7,876,264

Table 2.3: The number of concepts and relations in different ontologies compared in this study [61]

For the biomedical domain, we base our comparisons on two main existing references: McInnes et al. [126] and Garla et al. [61]. Both studies provide open-source software and perform the experiments on publicly available datasets. To compare against WordNet we used the baseline methods reported in [83, 1, 4]. We modified the disambiguated WordSimilarity353 for Wikipedia [202] to be used in this project. All biomedical datasets were semi-automatically disambiguated.

2.6.6 Comparison with the Relatedness Methods Based on Biomedical Ontologies

For the ontology-based methods, we base our comparisons on Garla et al. [61] which provides open-source software and performs the experiments on publicly available datasets

BenchmarkKB		Ontology						**Wikipedia**
		Path		Intrinsic IC [182]				
		WUP [205]	LCH [104]	Lin	Path	LCH	PPR [1]	**HITS-sim**
Pedersen	Wikipedia							**.71**
N=29	sct-umls	.49	.44	.45	.38	.38	.63	
	mesh-umls	.41	.42	.45	.44	.45	.16	
	umls	.70	.61	.72	.70	.70	.69	
Mayo	Wikipedia							***.52**
N=101	sct-umls	.05	.03	.09	.12	.3	.17	
	mesh-umls	.2	.26	.25	.25	.25	.05	
	umls	.38	.3	.39	.41	.44	.46	
UMN Sim	Wikipedia							†**.58**
	sct-umls	.21	.23	.22	.23	.36	.23	
N=566	mesh-umls	.26	.25	.3	.29	.29	.18	
	umls	.39	.4	.43	.43	.46	.41	
UMN Rel	Wikipedia							†**.51**
N=587	sct-umls	.14	.17	.16	.16	.3	.17	
	mesh-umls	.35	.34	.34	.34	.35	.18	
	umls	.32	.34	.35	.35	.39	.33	

Table 2.4: Comparison with ontology-based methods [61]: Correlation across measures and ground truth. * Significant difference with all MeSH-based and snomed-ct based methods ($p\text{-}value < .05$). † Significant difference with all methods ($p\text{-}value < .001$).

(another similar research providing the results for ontology-based methods is McInnes et al. [126], but Garla et al. provide better results, probably due to using different versions of the incorporated ontologies).

The best results belong to three methods: LCH (cf. Section 2.3.1), which is a path-based method; Intrinsic Information Content (IC) based LCH (IIC-LCH); which is the same as LCH but replaces the path with IC difference between the two concepts (cf. Section 2.3.1), and Personalized Pagerank-based algorithms (PPR) (cf. Section 2.3.2). Garla et al. [61] include these state-of-the-art methods in three experiments on three ontologies, SNOMED-CT, MeSH, and finally, all ontologies integrated in *umls*, forming a graph with around two million nodes and 7 million relations. We only include the highest Wikipedia results in this section (which were achieved by $HITS\text{-}sim$) in Table 2.4. It is noticeable that our Wikipedia-based method gives greater improvements on the bigger datasets. This table supports our initial claims regarding the suitability of Wikipedia, especially the results for the largest dataset (UMN relatedness), where our Wikipedia-based method outperforms all ontology-based methods by a wide and statistically significant margin ($p\text{-}value < .001$).

2.6.7 Comparison with Distributional Methods: Evaluating a Word2vec-MetaMap hybrid

The comparison with distributional methods is given in Table 2.5. The state-of-the-art corpus-based methods are Context Vector (cf. Section 2.3.3) and Tensor Encoding [194]. However, these methods are to some extent hybrid as they both use meta-thesauri to unify the text and map it to biomedical terms. Symonds et al. [194] report Tensor encoding results for the smaller test datasets only, while Context Vector is evaluated on larger datasets as well by Garla et al. [61], from which we report the results. We also report results using two versions of *word2vec*, one merely corpus-based and one hybrid version which uses MetaMap (cf. Section 2.5.1).

Method	Resources	Pedersen N=29	Mayo N=101	UMN sim. N=566	UMN rel. N=587
Vector	Mayo Corpus*+UMLS	.76	†.02	†.02	†-.13
Tensor	OHSUMED+UMLS	.76			
Word2vec	OHSUMED	†.34	†.26	†.36	†.29
Word2vec	OHSUMED+UMLS	**.80**	**.63**	†.39	†.39
HITS-sim	Wikipedia	.71	.52	**.58**	**.51**

Table 2.5: Comparison with distributional methods: Correlation across measures and ground truth. * Mayo Corpus of Clinical Notes. † Difference with $HITS\text{-}sim$ is significant ($p\text{-}value < .05$)

2.6.8 Evaluating *HITS-sim*: The Effect of Ordering

A comparison of our proposed method with other Wikipedia-based methods is shown in Table 2.6. We compare our method with WLM [202], which is the most popular structural method based on Normalized Google Distance [39] and with bibliometric graph similarity methods. From Distributional methods, we compared with CPRel [88] and ESA (cf. Section 2.3.3) (we report the results for both methods from [88]). Abstracting from the details, both methods generate a term-document matrix based on TFIDF. Given two terms, CPRel uses the Wikipedia pages associated with the terms and calculates the cosine similarity between the two document vectors from the term-document matrix, while ESA finds the correspondent rows for the terms in the term-document matrix and calculates the cosine between the two vectors (cf. Section 2.3.3). General terms are not well covered in

Wikipedia and the associated pages have a low-quality. This leads to inferior results with the structure-based methods. This will not affect ESA when dealing with general words (as ESA uses the text of Wikipedia as a corpus only), but on the other hand, ESA is not directly applicable to multi-word phrases (which is the case with most Wikipedia concepts). We used the same subset ($size = 318$) of WordSim353 used with CPRel.

It is observed that $HITS\text{-}sim$ is the only method that outperforms other methods on most of the test datasets. Relatedness test datasets are limited in size and this affects the significance of the differences. In Table 2.6, the significant differences with $HITS\text{-}sim$ are marked. Also following [3], we calculated the weighted average of correlations on WordSimilarity-353, Pedersen and UMN-relatedness (the three largest datasets with no overlap) and observed a significant difference between $HITS\text{-}sim$ and all structure-based methods (*WLM*, *co-citation*, *coupling*, *amsler*, SimRank and EHITS-sim) under $p\text{-}value < .05$.

Method	MC	WordSim353	Ped. Phys.	Ped. Coders	Ped. All	Mayo	UMN Sim.	UMN Rel.
ESA	.73	**.75**						
CPRel	.83	.64						
WLM†	.86	.67	.63	.69	.67	.49	**.58**	.49
Co-Citation†	.86	.67	.62	.68	.66	.47	.57	.49
Coupling†	**.90**	*.65	.61	.66	.64	*.44	*.49	*.4
Amsler†	.86	.68	.58	.66	.64	*.45	*.53	*.43
SimRank†	.79	*.51	*.56	*.55	*.55	*.39	*.45	*.39
EHITS-sim†	.84	*.62	.6	.67	.64	*.46	*.54	*.45
HITS-sim	.88	.70	**.67**	**.72**	**.71**	**.52**	**.58**	**.51**

Table 2.6: Comparison between Wikipedia-based methods: Correlation across measures and ground truth. * Difference with $HITS\text{-}sim$ is significant under $p\text{-}value < .05$ † Difference with $HITS\text{-}sim$ is significant on the weighted average of WordSim353, Ped. All and UMN Rel (three largest datasets that do not share any pair) under $p\text{-}value < .05$.

2.6.9 The Effect of Distance Method

A comparison of our proposed way of incorporating Kendall's tau distance with *cosine* metric as proposed in [117], is given in Table 2.7. Another measure that can take into account both importance and the ratio scale of the scores given by HITS, is *Pearson* correlation. The lower performance of both *cosine* and *Pearson* is because the compared scores

are the results of calculations performed on different graphs, in other words, the compared scores are in two different spaces.

Datasets	Kendall's tau(τ)	Pearson (r)	Cosine (cos)
Pedersen	.71	.57	.64
MayoSRS	.52	.42	.52
UMN Rel.	.58	.35	.55
UMN Sim.	.51	.36	.49

Table 2.7: The effect of the distance method used in Algorithm 1 for three distances: Kendall's tau (τ), Pearson (r) and Cosine distance (cos). Values are Spearman's correlation (ρ) with the gold standards.

2.7 Complexity Analysis

The similarity calculation should be as fast as possible in order to be useful in human interactive processes such as search engines, or in the inner loop of other computationally intensive algorithms such as clustering or classification. For Wikipedia-based methods reviewed in this project, one general rule of thumb is that methods working on outgoing links are preferable, because for most nodes we have $|O(v)| \ll |I(v)|$. Basic bibliometric methods and WLM only need to calculate the intersection of the two sets, which can be done with $O(n \times \log(n))$ operations w.r.t to the number of the neighbours. SimRank- based methods have a higher complexity, $O(n^3)$, where n is the size of the adjacency matrices.

Our proposed HITS-based algorithms only require the principal component and therefore, the *power method* can be used, which is very efficient with sparse matrices [68]. To give an intuition for the sparsity of the matrices we have included sparsity histograms in Figure 2.5 (sparsity is defined to be the proportion of zero elements). The histograms show that most matrices are sparse. It should be noted that calculating HITS for each concept is a one-time job; we run HITS offline and pre-compute the ranks of the neighbours for each node. Also Kendall-tau can be calculated more efficiently with $O(n \times \log(n))$ operations [38].

2.8 Conclusion

We gave a new comparison between different algorithms for Semantic Relatedness in the biomedical domain. We draw the following conclusions from our experiments:

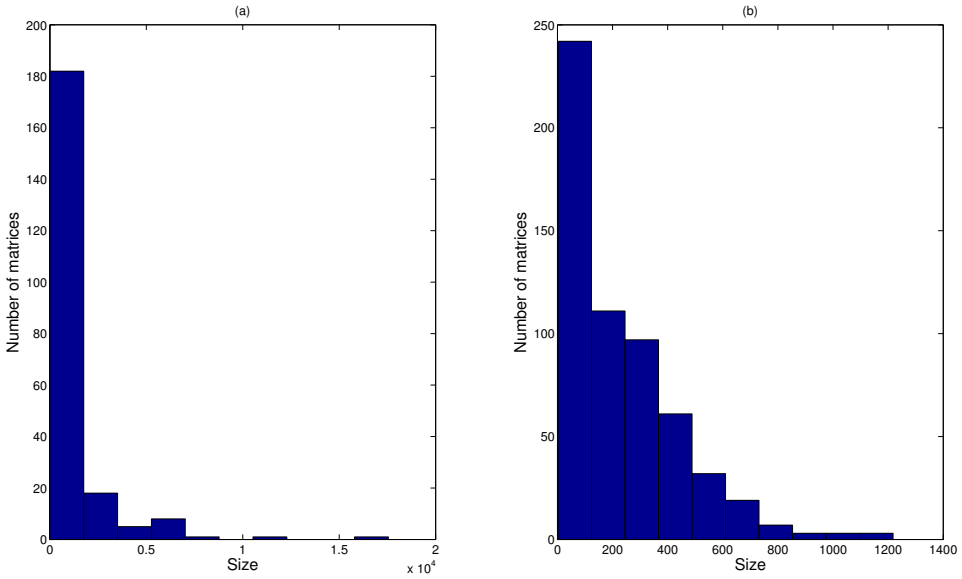


Figure 2.4: Distribution of the size of the matrices used in the experiments (a) in-neighbourhood graphs, (b) out-neighbourhood graphs.

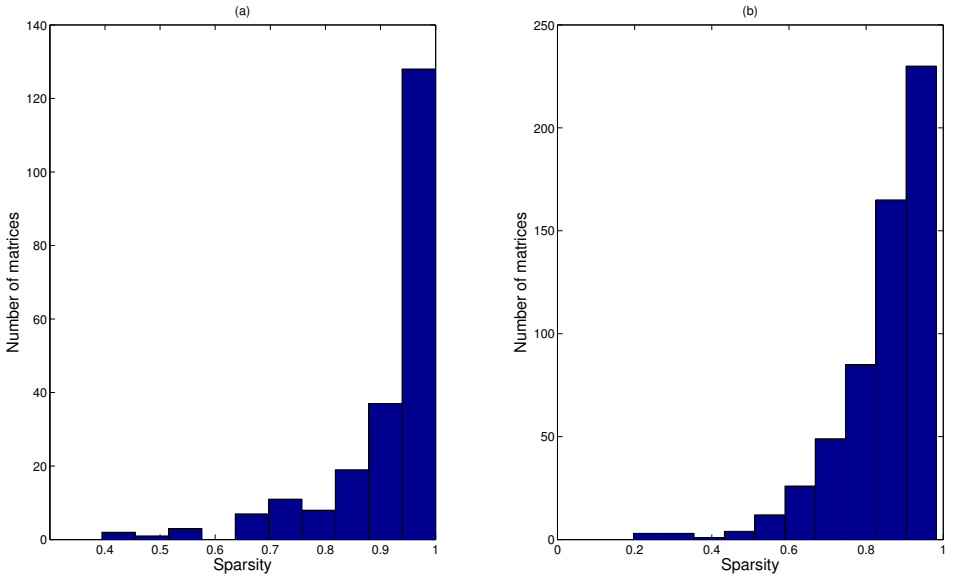


Figure 2.5: Distribution of the sparsity (the proportion of zeros) of the matrices used in the experiments: (a) in-neighbourhood graphs, (b) out-neighbourhood graphs

- We evaluated *word2vec* and *word2vec+umls* for biomedical domain for the first time. This demonstrated that distributional and ontology-based methods can be quite competitive and a hybrid of them improves the results.

- Wikipedia as a resource is comparable with the available specialized resources and often even significantly improves upon them (Tables 2.4 and 2.5).

- Our new proposed graph-based approach for computing relatedness based on the HITS algorithm achieves the best correlations with human judgement as illustrated in Table 2.6. We chose the biomedical domain because of the availability of different ontologies and methods, which is significantly higher for any other domain.

Chapter 3

Vector Space Representation of Wikipedia Concepts

3.1 Introduction

Semantic relatedness is a real-valued function defined over a set of concept pairs that can reflect any possible taxonomic or non-taxonomic relation between them. This measure can be extracted from either unstructured corpora or lexical resources, each with their own pros and cons [1]. Among knowledge-based methods, Wikipedia is gaining popularity due to its broad coverage of concepts and named entities in different domains; previous research shows that it can perform close to or even better than human curated domain-specific ontologies and corpora [181]. Wikipedia's graph structure provides a rich source for many graph based Natural Language Processing (NLP) methods and has been used extensively in text analysis [139, 70, 54]. Our research is motivated by this graph structure and investigates efficient and effective ways to represent a concept using this structure for calculating semantic relatedness. Relying on vector representations is not necessary for semantic relatedness calculation, but it can provide a large repository of methods and techniques, referred to as Vector Space Model (VSM).

The proposed method is in fact a compromise between two extremes: one is to use only in-coming or out-going links of a concept to represent it [202], and the other is to use the whole Wikipedia graph to extract the representation [209, 2]. While the former keeps the task simple, it does not take advantage of the full network structure, and the latter makes the task so complex that it is practically intractable. We conjecture that using the neighbourhood of a vertex benefits the representation compared to merely in or out-links, and also that using vertices further away does not contribute to the representation, and may even decrease the quality. We demonstrate the quality of the representations in the semantic relatedness task. Moreover, we report the successful incorporation of our concept representation in *query expansion* for a *microblog filtering* system [118, 119]. In this mentioned study, several state-of-the-art embedding methods are compared for *query expansion*, and our method obtains the best results.

3.2 Related Work

3.2.1 Vector Space Representation of Concepts

In computational linguistics and information retrieval, vector space representation of concepts is the dominant method of representation and a wide variety of methods are used to obtain this representation. These representations are either sparse or dense. Each of these approaches have their pros and cons, briefly explained in the following sections.

Sparse Representation

The simplest way to achieve term representation is using the rows of a *term-document* frequency matrix, or a *term-term* frequency matrix (a.k.a., co-occurrence matrix) [122, 13]. A *term-document matrix* is a matrix $\mathbf{X}$ with the terms corresponding to its rows and documents corresponding to its columns, and each entry X_{td} equals the TFIDF value of term t w.r.t document d, which is defined as

$$tfidf(t,d) = tf(t,d) \times idf(t), \tag{3.1}$$

where $tf(t,d)$ is the frequency of term t in d and $idf(t)$ is the *inverse document frequency*, that is $N/df(t)$, where N is the number of documents and $df(t)$ is the number of documents containing t.

Using the matrix representation, every term t can be represented by $\mathbf{X}_t$ where $\mathbf{X}_t$ is the t-th row of $\mathbf{X}$, and for terms s and t, the similarity between them will simply be the cosine similarity between $\mathbf{X}_s$ and $\mathbf{X}_t$.

Different variations of this basic idea can be adapted to structured knowledge sources. For example, in the biomedical domain and when using biomedical ontologies, a well known method is context vector [156]. It extracts the descriptions of the biomedical concepts; the vectors associated to the terms of the description are looked up in a co-occurrence matrix built from a big corpus; and finally the vectors are averaged to form the concept representation. As another example, for Wikipedia concepts, a successful method is Keyphrase Overlap Relatedness (KORE) [78], which uses the anchor-texts of a page to represent the concept, weighted by inverse document frequency and mutual information. One of the earliest and most cited representation methods for Wikipedia is Explicit Semantic Analysis (ESA) [56]. ESA is a very simple approach which uses the concept article text to extract

the term-document matrix, therefore representing every term as a vector of Wikipedia concepts.

As we will explain in the next section, there is a great deal of research on how to convert a sparse representation to a dense one. However, sparse representations also have some advantages [141]:

1. They are interpretable. The dimensions of dense representations do not have real-word correspondents and hence, lack "cognitive plausibility".

2. Moreover, in dense representation, unlike sparse representation, the individual dimensions do not show any clear correlation among similar or dissimilar concepts. For example, a very active dimension in a fruit (a dimension with a large magnitude) can be active in a very unrelated concept.

3. Economy of storage: It is unlikely that the same features are used to represent all concepts, which is the underlying assumption of dense representation and *latent space*. Some might need more and some less, and experiments in prototype theory support this idea [62].

Dense Representation

The idea behind these methods is to take a sparse representation and convert it to a dense representation (often referred to as an embedding). There are multiple motivations for a transfer from a sparse space to a dense space:

- Overcoming the high dimensionality problem: it is sometimes referred to as *curse of dimensionality*, and it causes several problems. Among them is the *Hughes Phenomenon* [82], which states that the accuracy of a pattern recognizer decreases with the number of the dimensions. Another problem is concentration of similarity scores: most points in the space become quasi-similar in a high-dimensional space [214].

- Sparse representation cannot handle *synonymy* and *polysemy* [122]: synonymy is when two different words refer to the same concept and polysemy is when one term can refer to multiple concepts.

The classic and traditional way of representing a concept in a dense space is called Latent Semantic Analysis (LSA) [101]. Assuming that $\mathbf{X}_{m \times n}$ is the term-document matrix with

rank r, it uses Singular Value Decomposition (SVD) to extract k dense dimensions from $\mathbf{X}$. The new space is defined by the k largest eigenvectors of $\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T\mathbf{X}$. Every term vector t can now be represented in this new $k-$dimensional space by a vector $\hat{t}_k$, which is called the *dense embedding* of t. *SVD* is based on optimizing the Frobenius norm of the difference between $\mathbf{X}$ and its rank-k approximation, $\tilde{\mathbf{X}}$. Replacing this objective function with a different one results in different embedding. The most famous ones are *word2vec* [134, 133] and *Glove* [158].

Graph Embedding

Concepts can have multiple relations in an ontology, forming a graph. Representing a concept using a graph is not a straightforward task. Moreover, calculating similarity, and generally computation with graphs is "computationally cumbersome" [201] and hence, transferring it to vector space is a common technique. This is usually due to the fact that the nodes of a graph do not contain any natural order [201]. Representing a graph in vector space is one of the popular approaches in tasks involving graph similarity [171, 189, 65, 64, 41]. In the domain of concept representation, the embedding methods can again be sparse or dense.

One of the first experiments on using Wikipedia graph structure is to represent a concept by its links, and set the value of each link in the vector to some value that expresses its relevance to the article. This approach is investigated in [202], and the value of each link is set to be

$$w(s \rightarrow t) = \begin{cases} log\left(\frac{|W|}{|T|}\right) & \text{if } s \in T \\ 0 & \text{otherwise,} \end{cases} \tag{3.2}$$

where T is the set of all articles that link to t, and W is the set of all articles. Therefore, this method gives less importance to links pointing to articles with many other in-links. This method is outperformed by the famous Wikipedia Linked-Based Measure (WLM) [202] method in the same paper.

Sunflower [114] is another system that uses Wikipedia category structure to represent a concept. Every concept in Wikipedia is assigned multiple categories, and categories themselves can have taxonomic relationship with each other. Sunflower represents a concept by

a vector of categories, and the importance of each category is determined by the distance of the category from the concept in the category graph and also the number of times the concept is associated with the category across multiple languages. Sunflower has been used in several projects, including the Tulip entity linker [114].

However, the more widely used method for extracting concept representations from a graph uses a particular form of random walk called Personalized Pagerank (PPR). The idea of Personalized Pagerank is to find the limiting probabilities of a walker on the graph customized for a specific node v. This approach was investigated in [83, 16], and Personalized Reverse Pagerank was firstly hypothesised to be a more suitable approach in [16] using some qualitative examples from Open Directory Project (ODP) [1]. Later studies showed that PPR yields excellent results on WordNet [83, 1] and on biomedical ontologies [61].

The success of PPR-based methods on ontologies motivated subsequent researchers to try it on Wikipedia. The first attempt was WikiWalk [209]. As explained before, to get the representation of a node v, this method runs PPR customized for node v. To understand customization, we need to remember that random walk models a walker that either walks randomly to a neighbour or jumps to another node according to a probability distribution on the nodes called the *teleport vector*. For original Pagerank, this distribution is uniform, meaning that it can jump to any node, but *Personalized Pagerank* can put more emphasis on specific nodes (topics) by giving higher probabilities to them, which is why it is also called Topic-Sensitive Pagerank. WikiWalk uses this idea to calculate similarity in three steps:

1. For each term, calculate its ESA vector (cf. Section 2.3.3).

2. Execute PPR using the ESA vector as its teleport vector.

3. Calculate the stationary distribution for each word.

WikiWalk was not very successful. However, a more recent approach, UKB [2], achieved a significant improvement over WikiWalk by using only reciprocal links. This is interesting from theoretical point of view, but it has some limitations in realistic applications: first, using the whole link structure of Wikipedia is practically impossible; second, most of the non-popular entities of Wikipedia do not have a significant number of reciprocal links.

[1] `https://web.archive.org/web/*/http://www.dmoz.org`, taken from Archive.org

The main difference between the mentioned Personalized (reverse) Pagerank-based approaches [83, 16, 1, 209, 2] and our method for calculating the representation is that they use a global Pagerank on the whole graph, but personalized for a target node. On the other side, we conjecture that the neighbourhood graph of a concept is its primary representation, and we can use any ranking algorithm, including *non-personalized* Pagerank and rvsPagerank, to represent this graph using a vector.

The concept of dense embedding for graphs is a very popular topic in pattern recognition [18, 201]. One of the first approaches to graph embedding was *structured data embedding* [24] and a later version of it, *TransE* [23]. TransE focuses on a knowledge base that represents subject-predicate-object relationship, or head-relationship-tail (h, l, t) in their terminology. The objective function tries to assign embeddings to each of the elements of triples such that the embedding of h added to the embedding of l is close to the embedding of t. This method was successfully used to embed WordNet and Freebase [2]. A closely related method is *Knowledge Base Embedding* [75], which uses a deeper neural network, called "Deep Structured Semantic Model" (DSSM), to embed Freebase as well.

However, more recent studies show that for ontologies, looking only at the few adjacent nodes and concepts is not enough for effective embedding, and further nodes should be also taken into account. A successful approach in this direction is *DeepWalk* [159], which performs a random walk to convert the graph to a sequence and later uses the conventional text embedding to embed the nodes. *DeepWalk* is evaluated on a few graphs, such as the YouTube user group graph. *Large-scale Information Network Embedding* (LINE) [195] takes a similar approach to embedding a portion of Wikipedia and some other social networks. A more recent approach that extends *TransE* to further nodes is [198] which uses a deep neural network to embed the first- order and second-order proximities in the graph.

3.2.2 Relatedness in the General Domain

Semantic relatedness approaches are categorized as distributional or knowledge based. The term *structure-based* can also be used to describe methods using only the structure of a knowledge base, typically via graph representation. WordNet [136] is the primary knowledge source for traditional semantic relatedness methods and can yield highly accurate results on classical datasets [1]. Corpus-based methods can produce competitive results

[2] `https://developers.google.com/freebase/`

using large datasets and computational resources [1, 133, 158].

3.2.3 Relatedness from Wikipedia

The broad coverage of domain-specific terminologies and named entities in Wikipedia has made it a highly popular knowledge source in recent years. A wide range of approaches have been used to calculate semantic relatedness from Wikipedia, including adaptations of ontology-based (WikiRelate [161]), *distributional* (Kore [78], CPRel [88]), *graph-based* (Wikipedia Link Measure (WLM)) [202], HITS-Sim [181]) or *hybrid* (WikiWalk [209]) methods. To the best of our knowledge, WLM is the most popular method in different applications of Wikipedia-based semantic relatedness, such as *Named Entity Recognition* [168, 75, 19, 105] and *Link Prediction* [200]. It is an application of *Kolmogorov complexity*-based similarity [39] to the link structure. A more recent graph based method is *UKB* [2], which, as explained before, uses *PPR* on the Wikipedia graph to extract concept representation.

Several distributional methods have been proposed for utilizing the text of Wikipedia, such as representing a concept using the anchor texts (CPRel [88]) or the keywords (Kore [78]) in the page associated with it. More recent techniques are mainly focused on applications of word2vec [134] on the Wikipedia text to represent the concepts [188].

3.3 Wikipedia Graph

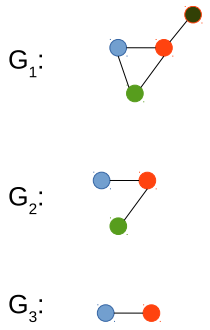
Wikipedia assigns a page to each concept, so a directed graph can be obtained with nodes representing concepts and edges representing out- links from one page to another. There exists a specific type of edge, called *redirect*. Redirecting denotes synonymy (for example *UK* is redirected to *United Kingdom*). We define the *Synonym Ring* of a node to be the set of synonyms and derive another graph that we refer to as *Wikipedia graph*, which results from grouping synonymous nodes to form a *meta* node and then merging the edges between nodes to obtain edges between meta nodes. A formal definition is given in Section 2.4.

3.4 Local Graph Embedding

Our approach for representing a concept in vector space is graph-based. Given a vertex v, the concept is represented in a space defined by *Wikipedia graph vertices*. This is done in

two steps:

- Step 1. Extract the closed neighbourhood graph for v, $G_v = (V_v, E_v)$ with adjacency matrix $\mathbf{A}_v$.
- Step 2. Embed G_v into a vector space defined by the Wikipedia graph vertices (Fig. 3.1).



E(G)				
$E(G_1)$	g_{11}	g_{12}	g_{13}	g_{14}
$E(G_2)$	0	g_{22}	g_{23}	g_{24}
$E(G_3)$	0	g_{32}	0	g_{34}

Figure 3.1: Illustration of the graph embedding process: each graph is mapped to a vector in a space defined by the Wikipedia concepts.

Five different methods are evaluated to embed the graph into vector space: Spectral embedding (Fiedler's vector) [47], HITS [98], Centrality (Katz) [93], and Pagerank [146] (and Reverse Pagerank [71]). All these methods return a normalized vector of assigned values to each node in V_v. This vector is then augmented by letting the value of vertices not in V_v to zero. This leads to the embedding of G_v, hence the representation of v, denoted by $\mathcal{R}(v)$.

We can use the whole neighbourhood, the in-neighbourhood or the out-neighbourhood to define a node's representation. We chose to embed both neighbourhoods separately and then calculate the average vector. This is different from embedding the whole neighborhood, since a node can be in both of the neighborhoods. But this way we can compare the importance of each neighbourhood as well. The results reported in this thesis are obtained by the average vector, unless otherwise stated.

3.4.1 Fiedler's Vector

Fiedler's vector [47] is the primary way of graph embedding and has been shown to be extremely successful in many different disciplines, from VLSI design and *finite element*

method to text clustering [193, 132]. This vector has many interesting properties, among them providing a balanced cut of a graph [193]. A natural embedding of a graph is to map it to a line, with the condition that the distances between neighbouring nodes should be small [192]:

$$\vec{x}^* = \arg\min_{\vec{x}} \sum_{(u,v)\in E} (x(u) - x(v))^2 \tag{3.3}$$

To avoid trivial answers like mapping everything to one single point, we need to satisfy the following conditions:

$$\begin{aligned} \sum_{u\in V} x(u)^2 &= \|\vec{x}\| = 1 \\ \sum_{u\in V} x(u) &= \vec{1}^T\vec{x} = 0 \end{aligned} \tag{3.4}$$

Applying these conditions to Eq. 3.3 leads to:

$$\vec{x}^* = \arg\min_{x\perp(1,1,\ldots,1)} \frac{\sum_{(u,v)\in E}(x(u) - x(v))^2}{\sum_{u\in V} x(u)^2} \tag{3.5}$$

If D is defined to be the diagonal degree matrix and A the adjacency matrix, the $Laplacian$ of an undirected graph is defined as

$$L = D - A \tag{3.6}$$

Using the *Laplacian*, we can rewrite the definition 3.3 as:

$$\vec{x}^* = \arg\min_{x\perp(1,1,\ldots,1)} \frac{\vec{x}^T L\vec{x}}{\vec{x}^T\vec{x}} \tag{3.7}$$

It can be proved [193] that $\vec{x}^*$, a.k.a. *Fiedler's vector*, is the *eigenvector* associated with the second smallest eigenvalue (λ_2) of the *Laplacian*. This value is also referred to as *Fiedler value* or *Algebraic connectivity* of the graph.

3.4.2 Hyperlink-Induced Topic Search (HITS)

Hyperlink-Induced Topic Search (HITS) and the notion of mutual reinforcement [98] is a well known concept in information retrieval, originally developed to rank a set of search results. HITS gives every node two scores: a *hub score* and an *authority score* through a recursion on the graph. So if it is run over a graph consisting of pages related to a concept

(*focused graph* in HITS terminology), the final product of the algorithm is two ranked lists associated to the focused graph: authoritative pages and those which are good hubs to the authoritative pages .

We are initially interested in finding a vector $\vec{x}$ which assigns a value called *authoritative value* to each node. But HITS is different than other embedding methods, in that the constraint ϕ on x is defined related to another embedding y, referred to as the *hub score*. The notion of *mutual reinforcement* denotes this simultaneous fulfilment of the two constraints. For each node u, let the *authority score* of u be $x^{\langle u \rangle}$, and the *hub score* be $y^{\langle u \rangle}$. Then the constraint is defined as:

1. The x score of nodes pointed to by nodes with higher y should be higher.

2. The y score of nodes pointing to nodes with higher x should be higher.

Assuming that E is the set of the edges, HITS initializes every score with 1 and performs the following iterations for each node p:

$$\phi : \begin{array}{rcl} x^{\langle u \rangle} & \leftarrow & \displaystyle\sum_{v:(v,u)\in E} y^{\langle v \rangle} \\ y^{\langle u \rangle} & \leftarrow & \displaystyle\sum_{v:(u,v)\in E} x^{\langle v \rangle} \end{array} \tag{3.8}$$

Or in matrix form:

$$x \leftarrow A^T y \tag{3.9}$$

$$y \leftarrow A^T x \tag{3.10}$$

By normalizing these scores after each step, assuming $\mathbf{A}$ is the adjacency matrix, it is provable that these equations converge and the final value of X, the vector of all x scores, will be the principal eigenvector of $A^T A$ and final value of Y, the vector of all y scores, will be the principal eigenvector of AA^T [98].

Extended HITS [20, 187] proposes calculating the principal eigenvector of $\mathbf{A}\mathbf{A}^T + \mathbf{A}^T\mathbf{A}$ to capture similarity between different nodes.

3.4.3 Katz Centrality

Katz centrality [93] has been shown to be most successful in link prediction [110], and for two given nodes, it is defined as the weighted sum of all paths lengths between them. Given

the neighbourhood of node v, we use the following vector to represent a graph:

$$\vec{katz}(i) = \sum_{l=1}^{\infty} \sum_{j=1}^{n} \alpha^l (\mathbf{A}^l)_{ji}, \tag{3.11}$$

where $\vec{katz}(i)$ is the Katz centrality of node i and α is a parameter with a value between 0 and 1. With one condition, α being smaller than the reciprocal of the absolute value of the largest eigenvalue of $\mathbf{A}$, $\vec{katz}$ can be calculated directly from:

$$\vec{katz} = ((\mathbf{I} - \alpha \mathbf{A}^T)^{-1} - \mathbf{I})\vec{1}. \tag{3.12}$$

3.4.4 Pagerank

Pagerank [146] is another link analysis algorithm primarily used to rank search engine results. It is defined as a process in which starting from a random node, a random walker moves to a random neighbour with probability α or jumps to a random vertex with the probability $1 - \alpha$. The Pagerank values are the limiting probabilities of finding a walker on each node.

Let $\mathbf{D}$ be the diagonal matrix with the out-degree of each node on the diagonal. If we set $\mathbf{W} = \mathbf{A}^T \mathbf{D}^{-1}$, then the Pagerank vector, initialized with $\vec{1}/n$, can be obtained from the following recursion:

$$\vec{pr}_{t+1} = (1 - \alpha)\frac{1}{n}\vec{1} + \alpha \mathbf{W} \vec{pr}_t. \tag{3.13}$$

It can be shown that the stationary probabilities can be calculated as

$$x = \frac{1 - \alpha}{n}(I - \alpha \mathbf{W})^{-1}\vec{1}. \tag{3.14}$$

3.4.5 Reverse Pagerank

Reverse Pagerank can be obtained from Pagerank simply by inverting the directions of the edges of the Graph. We had various motivations to analyse rvsPagerank and it showed to be the most successful embedding:

- HITS results showed that hub-scores were outperforming authority scores on out-neighbourhood, and rvsPagerank calculates the hub scores similar to HITS [51].

- It has a higher convergence rate and more potential to be locally approximated, i.e, the rvsPagerank of a target node can be approximated given only local information (neighborhood) [16].

- A Wikipedia page is defined and explained by its outgoing links rather than incoming links. In such cases, reversing the links makes more sense to get the importance of the page in a network, similar to the case of calculating trust [71].

3.5 Semantic Relatedness

Having the vector representations of two concepts, u and v, semantic relatedness is defined to be the cosine similarity of the two vectors, denoted by $\mathcal{R}(u) \cdot \mathcal{R}(v)$.

3.6 Alternative Approaches

In this section, we briefly explain other approaches we examined to calculate semantic relatedness from the Wikipedia graph structure. These methods can clarify some other aspects of the problem, and also can serve as powerful baselines for our experiments.

3.6.1 Global and Low-Dimensional Graph Embedding (Node Embedding)

A simple way to use a graph structure is to embed the whole Wikipedia Graph into a lower-dimensional space (around 300–500) and use the embedded vector assigned to each node as its representation, referred to as *node-embedding* in this research. We can achieve this by using a classical matrix factorization, such as SVD or Fiedler's vector, which we have discussed in several sections. However, our results using Fiedler's vector were not competitive.

We believe the problem with the classical spectral embedding is that it only tries to map adjacent nodes into a close proximity, while not having any explicit constraint regarding non-adjacent nodes. In other words, Eq. 3.3 tries to assign vectors to *Canada* and *maple* so that the Euclidean distance between them is minimized. However, by adapting the negative sampling approach [134], we can go one step further and try to keep the distance between *Canada* and *Hockey* minimum, while increasing the distance between *Canada* and a dissimilar word, say *banana*.

To include negative sampling in our formulation, we adapt the terminology of Levy & Goldberg [108] to our graph embedding problem. For every edge $(u, v) \in E$, two vectors are associated, $x(\vec{u})$ and $y(\vec{v})$. While assigning two vectors to each node is not motivated in the original paper, it has been explained in the literature [66, 172].

We try to convert the distance between the embeddings $x(\vec{u})$ and $y(\vec{v})$ to the probability of them being neighbours: $\sigma(x(\vec{u}) \cdot y(\vec{v}))$, where

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.15}$$

The log-likelihood method to calculate both $\vec{x}$ and $\vec{y}$, will result in

$$\theta^* = \arg\max_{\theta} \sum_{(u,v)\in E} \log \sigma(x(\vec{u}) \cdot y(\vec{v})) \tag{3.16}$$

where $x(\vec{u}) \in \mathbb{R}^d$; $y(\vec{v}) \in \mathbb{R}^d$, d is the dimensionality of the embedding and θ^*, the solution, includes both $\mathbf{x}^*$ and $\mathbf{y}^*$. Theoretically either $\mathbf{x}^*$, $\mathbf{y}^*$ or any combination of them can be used as the target embeddings, however usually $\mathbf{x}^*$ is used.

Now, we move forward to adding the negative sampling. We achieve this by defining another set E' as the complement of E, i.e, the set of all edges not in E:

$$E' = V \times V - E \tag{3.17}$$

We can modify Eq. 3.16 to include the negative examples as well:

$$\mathbf{x}^* = \arg\max_{\theta} \sum_{(u,v)\in E} \log \sigma(x(\vec{u}) \cdot y(\vec{v})) + \sum_{(u,v)\in E'} \log \sigma(-x(\vec{u}) \cdot y(\vec{v})) \tag{3.18}$$

However, in real applications, only a subsample of E' is used, constructed in a specific way: for every $(u, v) \in E$, add k samples $(u, v_1), \cdots, (u, v_k)$ from E'. Here, k is the number of negative samples, and we will show that increasing k will significantly improves the embedding results.

3.6.2 Graph Similarity Metrics

Vectorizing the two graphs is not the only way to compare them, because there exist (pure) graph similarity metrics. In our case, the two graphs are both induced sub-graphs of Wikipedia and therefore, vertex overlap would result in edge overlap to some extent, referred to by Graph-Overlap (or simply *overlap* in the tables). A better approach is to use a

variation of Normalized Graph Edit Distance (NGED) [58] to compare the graphs. NGED, similar to edit distance of strings, tries to convert one graph to another by means of a few operations (vertex/edge insertion/deletion/substitution) with different costs for each operation. The problem is NP-complete in general. In our case, edge substitution is never the case (they are induced subgraphs of the same graph). We let vertex substitution cost be ∞ for two vertices that have different labels and every other cost be 1, and derive a simple version of NGED:

$$NGED(G_1, G_2) = \frac{|V_1 \triangle V_2| + |E_1 \triangle E_2|}{|V_1 \cup V_2| + |E_1 \cup E_2|} \tag{3.19}$$

where $\triangle$ is the *symmetric difference* of the two sets.

3.7 The First Extrinsic Evaluation: Query Expansion

In this section, we briefly report another evaluation of our embedding vectors, which uses them in a *query expansion* task. These experiments were done by Makki et al. as part of their "Microblog Filtering System" project [118, 119]. The task is to design a *Twitter recommendation system* to retrieve an *ordered list* of novel and relevant tweets given user profiles. User interests are explicitly stated in textual format, as defined in TREC 2015 Microblog Track (Scenario B, also referred to as email digest) [113]. The solution proposed by Makki et al. has two phases: forming a query and retrieving the results. One of the main challenges is the inconsistency in the vocabulary, i.e, different terms are used to refer to the same, or closely related concepts. This type of inconsistency is often a clear case for query expansion. To expand a query, Makki et al. use semantic relatedness. Assuming that $\widetilde{Q} = \{q_1, \cdots, q_n\}$ is the initial query, a set of top m semantic related terms, s_i, is associated ($s_i = \{s_{i,1}, \cdots, s_{i,m}\}$) with each query term q_i, and Q, the final query, will be $\cup_i s_i$. Having constructed Q, the tweets can be retrieved using a standard information retrieval approach. However, due to the specific features of the task and evaluation method, several customizations and parameter tunings should be performed, including different ways of weighting named entities and non-entities in the query [118, 119].

Makki et al. examined several approaches to expand the query. However, we only focus on the winning strategy, which is a combination of *Named Entity Recognition* (NER) followed by an expansion strategy, which they refer to as $expand\text{-}common$. This approach is a direct way to deal with ambiguity: concepts can be ambiguous and expanding them can

further increase the ambiguity. In this strategy, top-N most related concepts are extracted for every term in the query, and then only those terms that appear in at least two of these expansions are added to the final query. Formally, $expansion(\widetilde{Q})$, the set of query terms that should be added to $\widetilde{Q}$ to form Q, is defined as

$$expansion(\widetilde{Q}) = \bigcup_{i,j=1,i\neq j}^{|\widetilde{Q}|} sem(q_i) \cap sem(q_j), \tag{3.20}$$

where $sem(q_i)$ is the set of top-N related concepts to q_i. As an example from the dataset, given a sentence like "U.S Forest Fires", the term "Wildfire" is added because it is in the top-N related words for both "Forest" and "Fire".

We need to emphasize that there is one important difference between the way top-N related terms are extracted in our framework (Wikisim), compared with other methods: because Wikisim is a concept representation in an explanatory space (dimensions are themselves concepts, and are therefore explainable), instead of finding N nearest neighbours, which is very expensive, Makki et al. used the top-N dimensions of the embedding (dimensions with the highest scores), which can be done in constant time.

3.8 Experiments

As pointed out by [181], Wikipedia does not cover general domain words as well as a general lexicon such as WordNet, while its main advantage is in domain-specific vocabularies and entities. We evaluate all methods on both general and domain-specific datasets but emphasize that due to the large number of non-covered or low-quality pages for the general domain, the comparisons on the domain-specific datasets (the right side of Table 3.1) are more meaningful.

Relatedness datasets typically are lists of word pairs along with their relatedness associated by experts. In the general domain, we use the following three datasets:

- Miller and Charles (MC) (28 pairs) [137]
- Rubenstein and Goodenough (RG) (65 pairs) [178]
- WordSimilarity353 collection [50] disambiguated to Wikipedia (318 pairs) [138]

For domain-specific datasets, we focus on the biomedical domain that has high quality datasets and are mapped to Wikipedia concepts [181]. We also evaluate on one dataset specifically designed for Wikipedia (*Kore-relatedness*):

- *Pedersen* (29 pairs) [156]
- *Mayo benchmark* (101 pairs) [148]
- *UMN Similarity benchmarks* (587 pairs) [147]
- *UMN Relatedness benchmarks* (566 pairs) [147].
- *Kore-relatedness* (400 pairs). [78]. This dataset has a different structure. The task is to sort a list of concepts with respect to their relatedness to a given target concept. The dataset includes 20 such lists.

3.8.1 Baselines

We use a range of methods from graph-based to distributional to evaluate our method:

Graph Based methods

Among graph based methods, we use WLM (cf. Section 2.3.2), UKB (cf. Section 3.2.1) [2], HITS-Sim (cf. Section 2.5.3) and two graph similarity metrics: overlap and NGED (cf. Section 3.6.2). We reimplemented WLM and for UKB, we used the provided source code [3].

Distributional methods

Among distributional methods, we chose the most recent approaches, CPRel (the cosine similarity between the Wikipedia pages) [88], Keyphrase Overlap Relatedness (KORE) (cf. Section 3.2.1) [78], Normalized Google Distance (NGD) [39] and the word2vec embedding method [134] (cf. Section 2.3.4). Concepts are often more than one single word and this affects word2vec performance, hence we tried two different approaches:

1. $word2vec_1$: Word2vec is shipped with a phrase detector [134]. It performs multiple initial scans over the text before it starts embedding, and in each scan, groups bigram

[3] http://ixa2.si.ehu.es/ukb/

phrases xy based on the following scoring formula:

$$score(x, y) = \frac{count(x, y) - \delta}{count(x) \times count(y)}; \tag{3.21}$$

δ is used as a threshold to avoid too many infrequent phrases.

2. $word2vec_2$. We use the embeddings from [188]. This approach goes through a preprocessing that uses the link structure of Wikipedia and unifies different entity mentions by replacing them with the entities they are linked to (best results were obtained with dimensionality set to 300).

3. NGD. *Normalized Google Distance* is a text-based and also a count-based similarity measure inspired by *Kolmogorov complexity* [39]. Given two terms x and y, it calculates similarity using the following equation:

$$ngd(x, y) = \frac{\log(\max(count(x), count(y))) - \log(count(x, y))}{\log(|V|) - \log(\min(count(x), count(y))} \tag{3.22}$$

where $|V|$ is the size of the vocabulary.

3.8.2 Parameters

We did not perform any parameter tuning and used popular default values for the constants, i.e, $\alpha = 0.005$ for Katz and $\alpha = 0.85$ for Pagerank. For word2vec based methods, we obtained the best results using 300 dimensions and 5 negative samples and a threshold $\delta = 10$ for phrases. Also for $node\text{-}embedding$, the model was trained with 500 dimensions and 20 negative samples. [4]

3.8.3 Relatedness Performance

The results provided in Table 3.1 show that neighbourhood embedding methods are mostly successful. More specifically, rvsPagerank stands out among all and obtains the most promising results. The only exception is UKB on RG and Pedersen, which are relatively small datasets compared to the rest. Moreover, UKB is more than 40 times slower than our method (1.74 sec per concept for rvsPagerank versus 78 sec for UKB).

[4] We use Wikipedia 20160305 dump for relatedness.

	General Datasets			Wikipedia Datasets				
Dataset	MC	RG	WS353	KORE-DS	Ped.	Mayo	UMN-sim	UMN-rel
Size	28	65	318	400	29	101	566	587
Text Based Methods								
$CPRel$	.83	.79	.64					
KORE (method)				.67				
NGD	.70	.78	.59	.0	.38	.14	.46	.44
$word2vec_1^\star$	.85	.77	.62	0	.35	.17	.17	.12
$word2vec_2^\dagger$	.81	.78	.63	.53	.66	.29	.30	.38
Structure-Based Methods								
WLM	.86	.82	.68	.68	.67	.49	.58	.5
UKB	.87	**.87**	.7	.66	**.82**	.39	.55	.51
HITS-Sim	.88	.81	.71	.67	.71	.52	.59	.52
$overlap$	.86	.8	.69	.63	.64	.44	.54	.44
NGED	.86	.79	.66	.6	.70	.48	.56	.5
Graph Embedding								
Node Embedding	82	78	.56	.69	.43	.35	.35	.28
Fiedler	.72	.7	.55	.52	.80	.5	.49	.44
Katz	.75	.73	.55	.62	.63	.52	.45	.37
HITS aut	.85	.72	.67	.56	.59	.49	.47	.41
HITS hub	.86	.77	.69	.62	.62	.52	.56	.51
Pagerank	.8	.69	.61	.54	.16	.12	.22	.15
rvsPagerank	**.90**	.82	**.72**	**.72**	.69	**.56**	**.62**	**.57**

Table 3.1: Comparison between Wikipedia-based methods: Correlation between measures and ground truth. *: automatic phrase detection, †: manual concept resolution using anchor texts

3.8.4 Global Graph Embedding

As we explained in Section 3.6.1, we are interested in the effect of negative sampling on the embedding as a differentiating factor from other similar methods. We evaluated our *node embedding* with different numbers of negative samples and report the results in Table 3.2. Increasing the negative sample number results in a significant improvement, although on some of the datasets the improvement is less noticeable.

	General Datasets			Wikipedia Datasets				
	MC	RG	WS353	KORE-DS	Ped.	Mayo	UMN-sim	UMN-rel
Size	28	65	318	400	29	101	566	587
# (Negative Samples)								
0	.30	.36	.02	.03	-.06	.03	-.02	-.02
5	.80	.74	.54	.60	.22	.15	.13	.05
10	.83	.79	.53	.68	.44	.30	.28	.20
20	.82	.78	.56	.69	.43	.35	.35	.28

Table 3.2: Comparison between different negative sampling numbers for *node embedding*: Correlation between measures and ground truth.

3.8.5 Which Neighbourhood Matters?

We had three options for the type of neighbourhood graph to use as the primary representation for the concepts: in-neighbourhood ($N_G^-[v]$), out-neighbourhood ($N_G^+[v]$) and the neighbourhood ($N_G[v]$) (formal definitions are provided in Section 2.4). In-links are preferred in most approaches [202] because they represent the traditional and widely used notion of *occurrence*: any occurrence of a concept c exhibits one inward link to the node associated with c. The other less popular option is the out-link set, which surprisingly, is the one that we are found more intriguing. The out-neighbourhood may be preferred for various reasons:

1. Having a significantly lower upper bound and a lower variance on the size (vertices having too many or too few neighbours): in the version of Wikipedia we experimented with, the maximum number of outgoing links is less than 3000, while the maximum in-link is almost one million (cf. Section 3.3).

2. The explained lower upper bound leads to a more robust and faster embedding on the out-neighbourhood. The non-linearity of the embedding calculation causes the out-going embedding to be much faster.

3. The high variance of in-neighbourhood size has a negative effect on the quality of the embeddings as well: around $275,000$ concepts do not have any in-coming links (this number is less than 1500 for outgoing links), leading to empty embeddings for all those concepts; on the other side, concepts with very high number of incoming-links will be represented very poorly, if at all.

We also tried to extract a neighbourhood using only *reciprocal links*, as suggested by Agirre et al. [2]. Similar to the other neighbourhood graphs, we can define the closed reciprocal-neighbourhood of a node ($N^c_G[v]$) as the subgraph induced by $I(v) \cap O(v)$.

As mentioned before, the problem with reciprocal links is, their very low frequency: many of the less popular concepts do not have any reciprocal links and by ignoring the non-reciprocal neighbours, we lose a large amount of information.

To use both *in* and *out*-neighbours, there are other options: either use the regular neighbourhood graph (this option is referred to as "ALL" in the tables), or separately embed *in* and *out*-neighbourhood graphs and take the average as the final vector (referred to as "AVG" in the tables).

We summarize the results in Table 3.4 and Fig. 3.2. The focus of Table 3.4 is on comparing the effect of embedding on different neighbourhoods. The first two rows in each group (*overlap* and NGED) are the non-embedding similarities, and rvsPagerank is our most successful method of embedding. We also include the results for Pagerank to understand the performance of rvsPagerank better. A more detailed view of the different embeddings for different neighbourhood graphs is presented in Fig. 3.2. We can draw the following conclusions from these results:

- We observe that, as is preferred, out-neighbourhood can benefit more from embedding and generate promising results.

- Embedding the graph has the least effect on reciprocal neighbourhood.

- The main advantage of rvsPagerank is its ability to embed out-neighbourhood while Pagerank performs very poorly, as expected.

- Except for *hitso* and *rvsPagerank*, most methods perform better on in-neighbourhood. The successful performance of $hitso$ was an indicator of the significance of hub-scores and was our main motivation to try *rvsPagerank*.

- Pagerank performs significantly better on reciprocal neighbourhood, which is consistent with the results of Agirre et al. [2].

- Averaging the in-neighbourhood and out-neighbourhood embeddings results in a better embedding compared with embedding the neighbourhood graph as a whole.

Our main focus in this section was on the graph structure. However, it can be noted that the main advantage of word2vec and generally distributional methods is their ability to represent non-concept words; they underperform when dealing with concepts, mostly due to the rarity of many concepts in the Wikipedia text, while on the other hand the graph is rich: many concepts are not mentioned in the text of Wikipedia, while they play a role in the graph structure via their links to other pages and categories.

Graph Direction	Property	Stats
IN	Size	1440
	Sparsity	.87
	T (on the fly)	1.74 (s)
	T (pre-embedding)	0.008 (s)
OUT	Size	302
	Sparsity	.78
	Time (s) (on the fly)	0.22
	Time (s) (pre-embedding)	0.001

Table 3.3: Graph statistics for different *in* and *out* neighbourhood graphs

3.8.6 Distance Metric: Do The Actual Values Matter?

In Chapter 2, we realized that when embedding a neighbourhood using HITS, what matters is the rank of the nodes and not the actual values associated to the nodes. In Fig 3.3, we compared the performance of *cosine* similarity vs our *top-K Kendall's tau* with several embeddings. Based on these results, we can conclude that the majority of methods perform better with Kendall's tau, especially Pagerank. The most interesting observation for us was the exceptional case of rvsPagerank, where *cosine* outperforms the rank-based metric on

		General Datasets			**Wikipedia Datasets**				
	Dataset	MC	RG	WS353	KORE-DS	Ped.	Mayo	UMN-sim	UMN-rel
	Size	28	65	318	400	29	101	566	587
Dir	Method								
IN									
	Overlap	.86	.82	.66	.64	.66	.47	.58	.50
	NGED	.85	.80	.66	.61	**.73**	.49	.59	.53
	Pagerank	.86	.83	.69	.68	.69	.52	.59	.53
	rvsPagerank	.84	.80	.65	.64	.68	.58	.59	.53
OUT									
	Overlap	**.90**	.75	.67	.63	.64	.44	.50	.41
	NGED	**.90**	.73	.67	.62	.66	.48	.51	.44
	Pagerank	.74	.54	.51	.28	-.11	.02	.14	.06
	rvsPagerank	.88	.80	.71	.71	.68	.53	.59	.53
REC*									
	Overlap	.85	**.84**	.68	.69	.66	.43	.53	.47
	NGED	.86	**.84**	.68	.69	.68	.45	.53	.46
	Pagerank	.83	.83	.68	.70	.63	.44	.52	.46
	rvsPagerank	.83	.83	.68	.70	.63	.44	.52	.46
ALL									
	Overlap	.86	.80	.66	.64	.64	.45	.54	.44
	NGED	.86	.79	.66	.60	.70	.48	.56	.50
	Pagerank	.74	.60	.53	.38	-.13	.02	.13	.05
	rvsPagerank	.85	.80	.67	.65	.69	**.59**	**.62**	**.57**
AVG†	rvsPagerank	**.90**	.82	**.72**	**.72**	.69	.56	**.62**	**.57**

Table 3.4: Comparing the quality of different neighbourhood graphs by evaluating their performance in embedding vs no-embedding evaluations. *: Reciprocal neighbourhood † Quoted Table 3.1

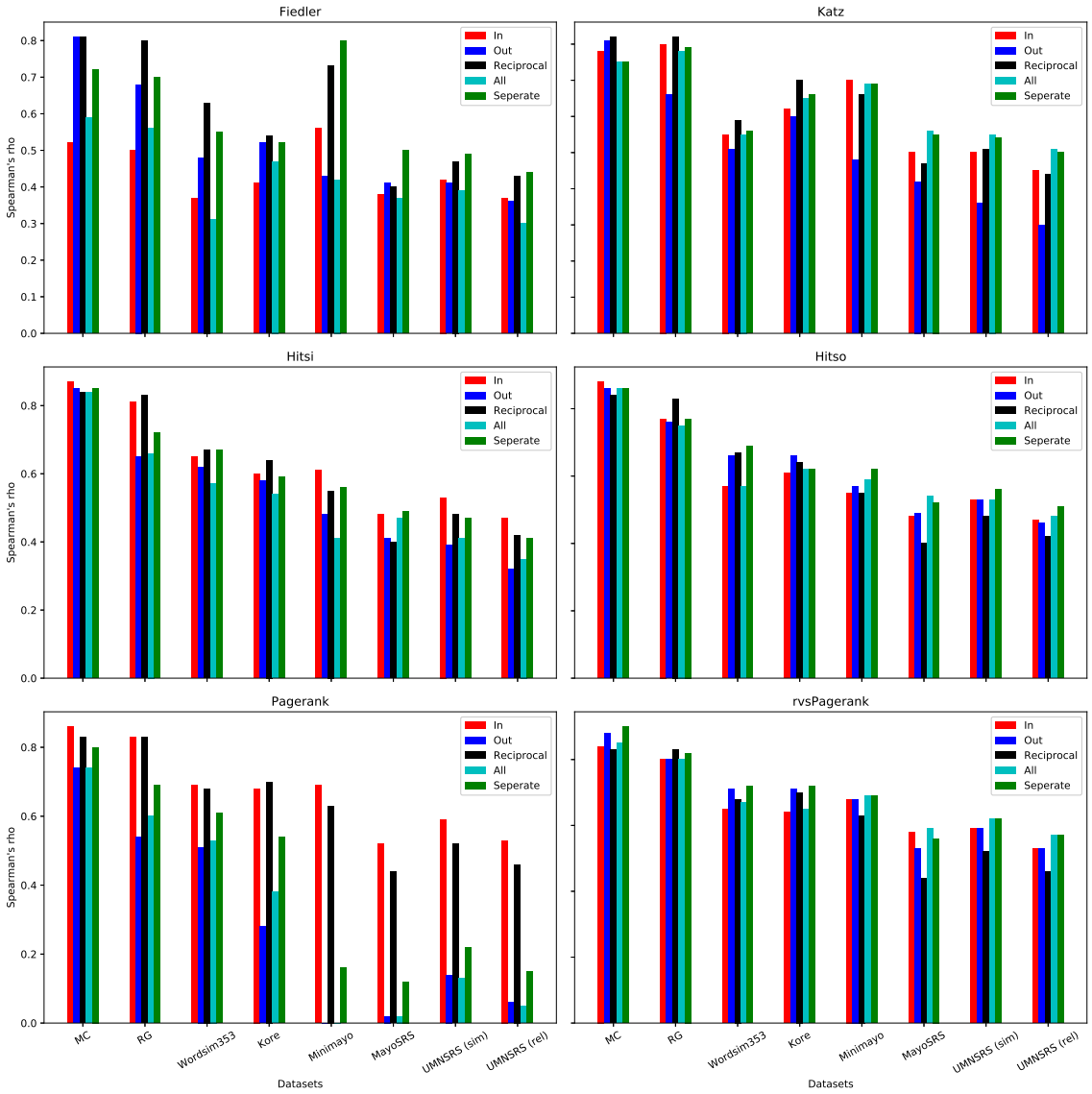


Figure 3.2: Comparison of the performance of different embeddings on different neighbourhoods: in-neighbourhood, out-neighbourhood, reciprocal, all neighbourhood and separate embedding and averaging.

all datasets. This can be a result of another feature of *rvsPagerank*, namely, its ability to approximate the global vector using only a local neighbourhood [16], as mentioned before.

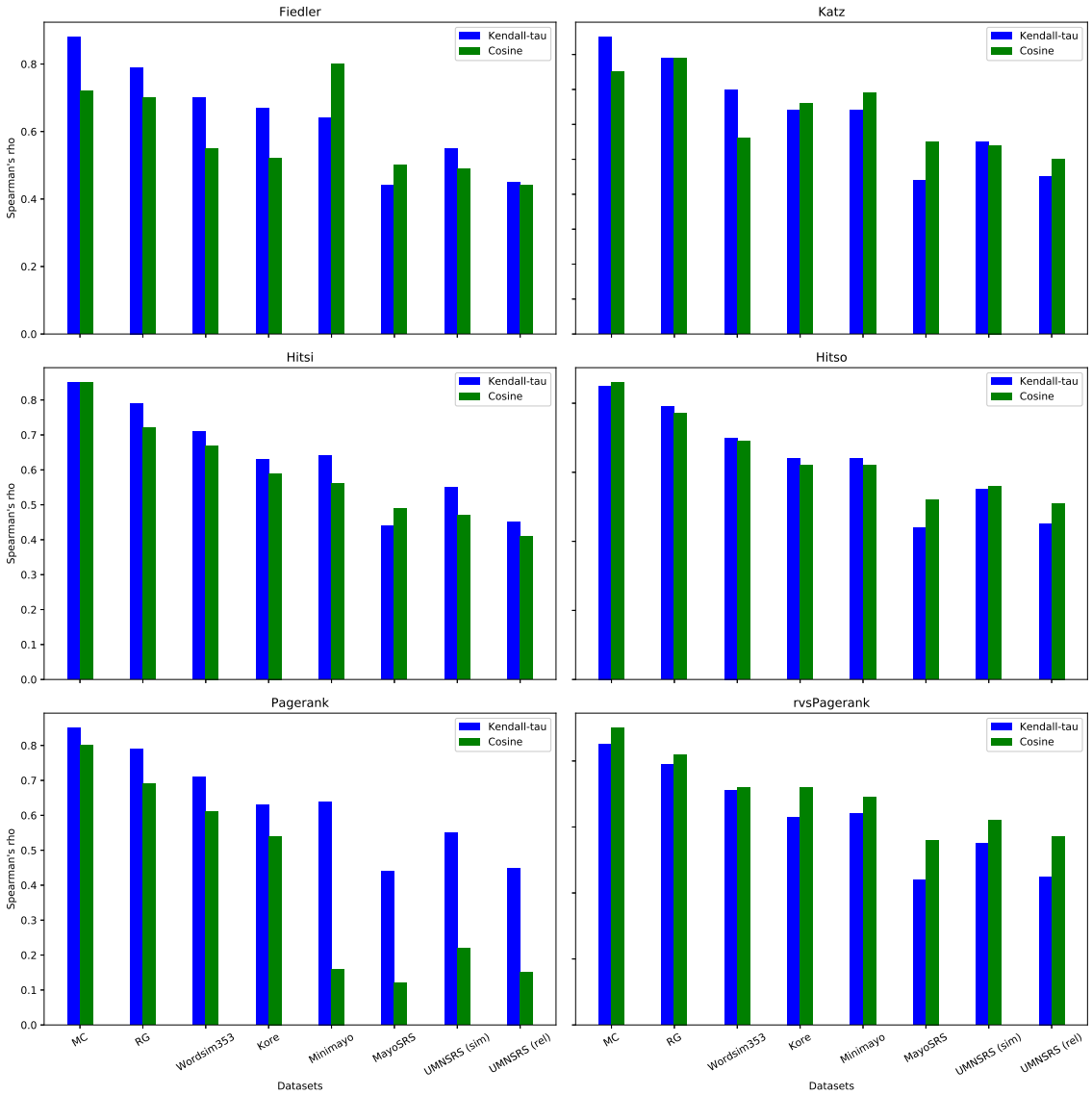


Figure 3.3: Comparison of the performance of different embeddings in semantic relatedness, across different datasets and using different metrics to calculate similarities between the embeddings

3.8.7 Off-the-Shelf Usage: Publicly Available Embeddings

Pagerank can be calculated using power iteration and can converge in a reasonable time. Table 3.4 summarizes some performance statistics about the method. While on-the-fly performance is acceptable (1.74 sec, compared to 78 sec for UKB), real-time performance can be achieved by calculating the embeddings offline. All the pre-embeddings, the source code for calculating the embeddings and the evaluated datasets, along with a web-service to facilitate the incorporation are available from the project website on MIT license[5]. Experiments are performed on a 32x2.0 GHz Intel core computer with 256GB of RAM.

3.8.8 Query Expansion

We report directly from [118, 119], and we only report the winning strategy among several approaches that are used for query expansion. Also, our focus is on the performance of different semantic relatedness methods in the query expansion, but we need to mention that none of the methods could outperform the baseline on *all* of the metrics. In this research, our proposed embedding (Wikisim) is compared with three other semantic relatedness methods: word2vec [134], GloVe [158] and UMBC [72]. UMBC provides a web service for a hybrid relatedness calculation method that uses LSA to obtain a low-dimensional representation, boosted with WordNet.

The dataset used in these experiments is a collection of over 16 million tweets, with a subset of them labelled according to the profile of 51 users [113]. The evaluation metric reported is *normalized Discounted Cumulative Gain* (NDCG), which is a widely used measure for the quality of a ranking in information retrieval [122]. Given the results of a search engine, i.e, a ranked list of size k with the ith element having a relevance score rel_i, $DCG@k$ is defined as $DCG@k = \sum_{i=1}^{k} \frac{2^{rel_i}-1}{log_2(i+1)}$, and $nDCG@k$ is defined as $nDCG@k = \frac{DCG@k}{IDCG@k}$, where $IDCG@k$ is the $DCG@k$ of the ideal ranking. However, two versions of this metric are used in the evaluations of this specific problem, based on how to score the systems on days that there exist no relevant tweets: $nDCG$-1, which assigns 1 to systems that correctly identify silent days, and 0 otherwise, and $nDCG$-0, which assigns 0 to all of the systems on silent days. The reported values are averaged over the evaluation period, and over different profiles, for $k = 10$ and $k = 5$. Makki et al. also

[5] `https://github.com/asajadi/wikisim`

report Mean Averaging Precision (MAP-1) over the top-100 results, with the same policy for silent days as $nDCG$-1.

The reported results for *word2vec* were obtained using the vectors on the Google News dataset. Trained versions on the Twitter dataset and with different dimensionalities were tested, but the results were not significantly different. Also, in the case of GloVe, pre-trained vectors on Wikipedia 2014 and English Gigaword Text [151] (resulting in very large dataset with 6 Billion tokens) are used. The performances of Wikisim and other relatedness methods are reported in Table 3.5. The queries are expanded with the top-N related terms ($N = 10$). Wikisim outperforms state-of-the-art semantic relatedness methods on all of the metrics. This extrinsic evaluation provides more basis for the quality of the vectors.

Method	$nDCG$-1		$nDCG$-0		
	@10	@5	@10	@5	MAP-1
UMBC	.3274	.3265	.1313	.1304	.2802
word2vec	.3555	.3599	.1437	.1479	.2997
GloVe	.2874	.2906	.1187	.1220	.2445
WikiSim	**.3595**	**.3646**	**.1470**	**.1508**	**.3029**

Table 3.5: Results when combining a Twitter filtering system with different semantic relatedness methods for query expansion, using nDCG@10,@5 and MAP-1

3.9 Conclusion

We presented a vector space representation for Wikipedia concepts. The representation is constructed by vector space embedding of the neighbourhood graph of the concept. We compared our representations with several state-of-the-art structure-based and corpus-based methods, and demonstrated that our method can outperform similar methods on various relatedness datasets. We also tried a wide range of alternative methods that were explored on Wikipedia for the first time, such as graph-based similarities, Normalized Google Distance, and comparing global and local embeddings. We showed that while global embedding is outperformed by local embedding, *negative sampling* is a key factor in the quality of the global embedding and can improve the results significantly. We also compared different ways of incorporating Wikipedia hyperlink structure and concluded that outgoing links carry more information than incoming links, while needing a significantly shorter

computation time to be processed. We also reported another experiment where our representations were used for *query expansion* in a *microblog filtering* competition [118, 119]. Our representation performed better than the top embedding methods.

Chapter 4

Word Sense Disambiguation

4.1 Introduction

WSD is a classical NLP task with a wide range of approaches. The problem is to assign the correct sense to a word that can have multiple *meanings*. For example, the term *bank* can refer to either a *financial institution* or *sloping land*. These two different meanings are said to be *homonyms* and are called multiple *senses* of the word *bank* [91].

In the context of Wikipedia, WSD is defined as a subtask of a more general process called *Named Entity Linking* (NEL) or *Named Entity Normalization* (NEN), that is linking mentions of entities in the text to a knowledge base. For example, given a sentence such as "David started dating Victoria, after she attended a Manchester United match", it is able to *recognize* (or *detect*) "David", "Victoria" and "Manchester United" and *disambiguate* (*link*, *normalize*) them to "David Beckham","Victoria Beckham" and "Manchester United F.C.", respectively.

WSD is conjectured to be AI-complete [120, 143]. A problem is AI-complete, by analogy with NP-complete (but not as mathematically rigorous), if solving it is the equivalent of solving the *artificial intelligence* problem, i.e., creating *human-level intelligence*. It has numerous applications, such as machine translation, information retrieval, speech processing and text spelling processing [85].

To evaluate our vectors in WSD, we start from the standard *coherence model* that finds a set of entities that maximizes *coherence*, defined to be the sum of pairwise similarities. This approach provides a suitable way to evaluate our relatedness method because the similarity can be any method that simply assigns a real value to a pair of entities. Being an *NP-complete* problem [100], one option is to formulate it in Integer Programming to provide a fair comparison with the most popular relatedness method in WSD, i.e, WLM [202]. Our main conjecture is that every short sentence contains one *key entity* that suffices to disambiguate the other entities. Using this assumption, we first simplify the *coherence*

definition and provide a quadratic-time algorithm that can confirm our conjecture experimentally. Using this finding, we provide a simple and linear-time complexity algorithm that can benefit from vector space calculations and achieve superior results compared to the coherence model, but with a dramatically lower cost.

4.2 Related Work

WSD approaches are classified as either distributional or LKR-based. Distributional methods can be further supervised or unsupervised according to the way they access the corpus [143], while the same logic can be extended to Lexical Knowledge Resource-based (LKR-based) methods, some classical resources assume that LKR methods are those that do not use any evidence from the corpus [5].

4.2.1 Unsupervised Methods

Any method that uses clustering to build word senses is called an *unsupervised method*. In the majority of these methods, different *contexts* in which a word appears are represented by feature vectors. These feature vectors are then given to a clusterer to generate a collection of several clusters, each representing a single sense. The feature vector for a given word w, i.e, its *vector representation*, is usually a simple co-occurrence vector where each element of it represents one word that co-occurs with w ([184, 155]). These vectors are sometimes called *first order context vectors*, from which *second order context vectors* can be extracted by averaging the first order vectors of the context words [165]. More recent approaches use *neural embedding* (neural network based embeddings) to represent a word and its context [92, 212, 157]. Graph-based approaches can also be applied to differentiate senses of a word. In this class of methods, a *word graph* is built using co-occurrence (or any other type of relation). It is conjectured that the *hubs* in the graph represent the word senses [143], which can be revealed by either a simple iterative algorithm [197] or Pagerank [7]. Unsupervised methods have a higher flexibility, especially in the case of low-resource languages. However, intrinsic evaluation of unsupervised methods is not simple. Some of the possible methods to evaluate unsupervised methods are a technique known as *pseudo-words* [207], closely studying the results [143] or *automatic optimal assignment* of senses [154].

4.2.2 Supervised Methods

The most widely used disambiguation method is supervised learning on a labelled corpus. In this approach, WSD becomes a classification problem with the *context representation* of a given word as the feature set and the correct sense of the word as the label [91, 122]. The features can be extended to include more information, such as Part Of Speech (POS) or even *syntacto-semantic* features. Labelled data are annotated according to well-known dictionaries (a.k.a. *sense repositories* in this task), such as Longman Dictionary English [163] or WordNet [136]. A wide range of supervised and semi-supervised algorithms have been used in this task [125, 143]. SVM is reported to be the most successful learning algorithm by different studies [143, 106], and "*It Makes Sense*" (IMS) [213] is an example of a high-quality WSD. IMS uses a feature vector containing several pieces of contextual and syntactic information, and SVM as its learning algorithm. Similar to the case of unsupervised methods, modern supervised methods are taking advantage of *neural embedding* to represent words and contexts [199, 84], and also to model the classification problem [210]. There exist several corpora for training and evaluation of these systems, the most widely used ones are the SemEval datasets (for more details, cf. [150, 143]).

4.2.3 Knowledge-Based Methods

Another form of supervision, especially useful when there is a lack of training data, is *knowledge-based* word sense disambiguation. This is also the main approach for domain-specific word sense disambiguation. One class of these approaches is called *Lesk Like* methods, which is a family of methods in which a profile is created for each word using the knowledge base, and then the features of the ambiguous word are compared to these profiles. The original Lesk method [107] is a dictionary-based method that uses the definitions of the words in the dictionary as their profiles. Then it tries to optimize the maximum overlap between several senses, similar to what a *coherence model* does (cf. Section 4.4). Assuming that the given sentence has only two mentions m_1 and m_2, each with k_1 and k_2 candidates respectively, Lesk will pick a pair of candidates that have the maximum overlap in their dictionary definition, using $k_1 k_2$ operations [130]. The *simplified Lesk* algorithm, which is a more popular and even more successful variation [196], removes the combinatorial part of the algorithm and resolves each word individually. In this approach, the context of m_1, for example, is compared to each of the candidate definitions and the one

with the maximum overlap is selected. The Lesk method can be improved in the presence of labelled data by extending the profile for a word with all the contexts in which it has appeared [95, 196]. This method has the best performance among different variations of Lesk [91]. SenseRelate [152] is an extension of another variation of Lesk algorithm (called Adapted Lesk Algorithm [14]), that can perform word sense disambiguation using any semantic relatedness method. SenseRelate has been applied in the biomedical domain in several studies [60, 127].

Graph-based methods are also another way to approach word sense disambiguation. There are various ways to build a graph from an ambiguous sentence, and different graph algorithms to apply. A successful example is a method developed by Navigli and Lapata [144], which builds a graph by connecting the words in the sentence to their senses from WordNet and then expanding them by a depth-first search. The disambiguation is performed by ranking the nodes according to their importance, which is measured by their *connectivity*. A variation of this algorithm is a *random walk*-based method that for every ambiguous node v, runs a Personalize Pagerank from v and uses the final distribution over the candidates to choose the correct senses [8]. This latter algorithm has been applied to UMLS ontologies for the biomedical domain [9] and more recently to Wikipedia [2]. An extract of the WordNet graph for disambiguating *coach* in the ambiguous sentence "*Our fleet comprises coaches from 35 to 58 seats*" is illustrated in Fig. 4.1 [6]. Performing *Personalized Pagerank* on this graph identifies $coach^5$ as the correct sense for *coach*.

While using semantic relatedness is not necessary for WSD [103], most entity linking systems benefit from it. Given a sentence, a real value is defined to represent the coherence of its meaning, referred to as *semantic coherence*. This value can be used for disambiguation in a variety of ways, for example as a link weight in a subgraph mining algorithm [78], but mostly as a feature fed to a classifier [168, 37]. Most of the mentioned methods use WLM as a popular relatedness method, although there exist some other relatedness methods tailored especially for this task, such as Kore [78], or learned measures for WSD using *learning to rank* [31] or *convolutional neural networks* (CNN) [53].

4.2.4 Integer Linear Programming

The Entity Linking problem can be considered as a special case of the *Metric Labelling Problem* [96, 97], where the goal is to label n objects while maximizing the similarities

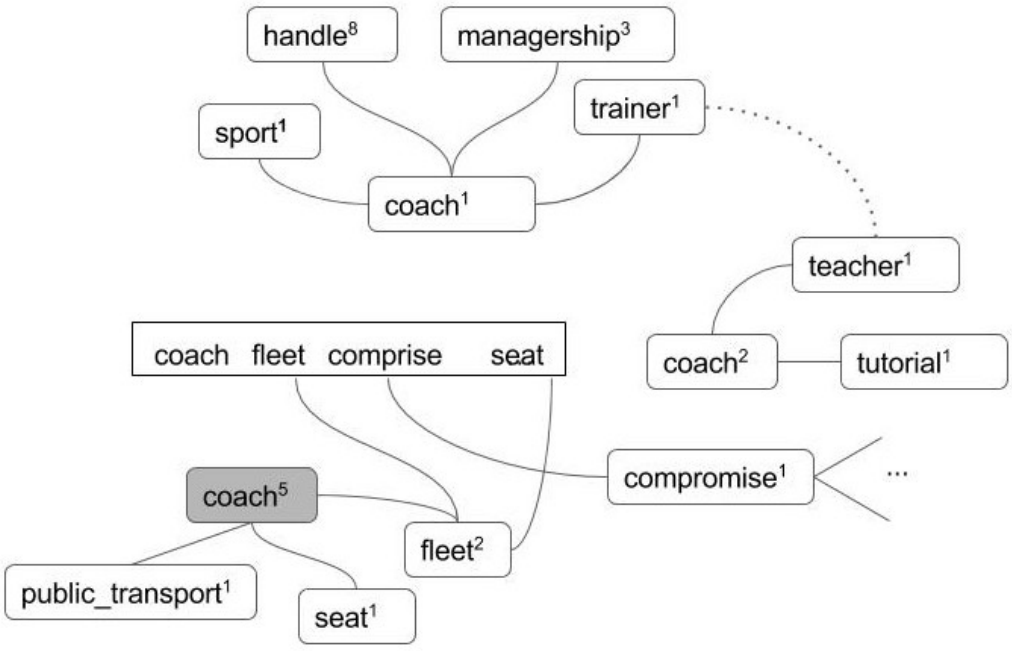


Figure 4.1: WordNet graph for disambiguating the ambiguous word *coach* in *Our fleet comprises coaches from 35 to 58 seats* [6]

between the related ones. The problem can be tackled using Integer Linear Programming [34]. In natural language processing, this motivated introducing a more general framework called Constrained Conditional Model (CCM), developed mainly by Roth et al. and discussed in several tutorials and papers [177, 176, 33, 67, 32, 36]. This framework has been used successfully in several NLP applications that involve *learning to label* objects while at the same time satisfying interdependent constraints. Examples are semantic role labelling [164], entity and relation identification [176] and wikification [168].

Linear Programming (LP) is an optimization method in which all the objective functions and constraints are linear. LP is a widely used optimization method in operations research. If all the variables are integers, the problem is called Integer Linear Programming (ILP), or often Integer Programming (IP). In this research, we start from a semantic relatedness-agnostic model for word sense disambiguation and transform it to an *integer linear program*. To solve the optimization, we use COIN-OR Branch and Cut Solution (CBC) [116]. Arbitrary sized sentences are chunked and fed to the solver to keep the problem tractable. While the details of solving an IP is not the main focus of this research, in a nutshell, CBC performs the following steps to solve a *minimization* optimization [52], with

integer variables allowed to take the values 0, 1 or 2:

- Step 1. (Bound): Relax the integral constraints, letting the variables take real values (with lower and upper bounds of 0.0 and 2.0 respectively). This new LP can be solved using classical linear programming solvers [142]. When solved, if all the variables have integral values in the obtained solution, we are done. Otherwise, this step will reveal a lower bound on the original problem. Any solution (an assignment that satisfies all the constraints) for the original integer problem is in fact an upper bound for the optimization.

- Step 2 (Branch). Pick a variable with non-integral value (e.g, 1.6), and branch two nodes from it, one labelled with an upper bound of 1 and the other labelled with the lower bound of 2

- While (search tree is not empty):

  - Step 3. Pick a node
  - Step 4. Create an LP program using the new lower and upper bounds, then solve
  - Step 5 (Bound). Try to prune the node if any of the following conditions hold:
    1. LP is infeasible
    2. Else, the value assigned to it exceeds its upper bound
    3. Else, if in the current solution all variables have integral values, update the upper bound, and prune the node by optimality
  - Step 6 (Branch). If the node was not pruned, pick another node with a non-integral value and branch.

4.3 Problem Definition and Formulation

In this section, we formulate the WSD problem as an optimization that merely is focused on the quality of the concept representations. We do this by ignoring other aspects of the problem, such as the effect of non-mention words, prior distributions of entities or null-mentions.

In the framework that we follow in this chapter, three steps need to be taken to disambiguate the words in a given input, as illustrated in Fig. 4.2:

1. Step 1. Mention Detection (a.k.a. Entity Recognition): This step aims to find any possible entities, or mentions of the knowledge base items. This problem is traditionally solved with Aho-Corasick algorithm [10], which identifies strings in a dictionary that appear in an input.

2. Step2. Candidate Generation: Every mention can potentially be linked to many entities. This might not be a problem in traditional WSD, but in linking to a knowledge base like Wikipedia, the number of possible targets can be very large and an early pruning can significantly affect the results.

3. Step 3. Disambiguation: This step selects the correct sense among each candidate set.

For every entity e, there exist one or more string representations, a.k.a. *mentions* of e, and vice versa. This forms a many-to-many relationship from mentions to entities. By ignoring non-mention phrases in the text, i.e, phrases that cannot possibly refer to any entity, we can represent a sentence S by a list of mentions $\mathcal{M} = [m_1, \ldots, m_n]$. Each mention m_i can have k_i potential senses or candidates: $\mathcal{C}_i = \{c_i^1, \ldots, c_i^{k_i}\}$ and $\mathcal{C} = \bigcup_i \mathcal{C}_i$ is the set of all candidates. The goal is to find $\mathcal{E} = [e_1, \ldots, e_n]$ where e_i is the "*correct*" entity to which mention m_i refers, using "*only*" the semantic relatedness between the concepts. Fig. 4.3 illustrate the notation we use in our formulation.

4.4 Coherence Modelling using Integer Programming (IP)

While there can be many different ways to measure the coherence of a sentence, the most popular one is defined by the sum of all mutual semantic relatedness scores of the entities of a sentence [100, 168]. We sometimes refer to this value as *standard coherence* measure. Let $r(\cdot, \cdot)$ be the relatedness function and $\hat{\mathcal{E}} = [\hat{e}_1, ..., \hat{e}_n]$ be a solution, i.e, each m_i is resolved to $\hat{e}_i$, then the problem is to find the vector $\mathcal{E}^*$ that maximizes the coherence:

$$\mathcal{E}^* = \underset{\hat{\mathcal{E}} \in (\mathcal{C}_1 \times \cdots \times \mathcal{C}_n)}{\arg\max} \sum_{i<j} r(\hat{e}_i, \hat{e}_j) \tag{4.1}$$

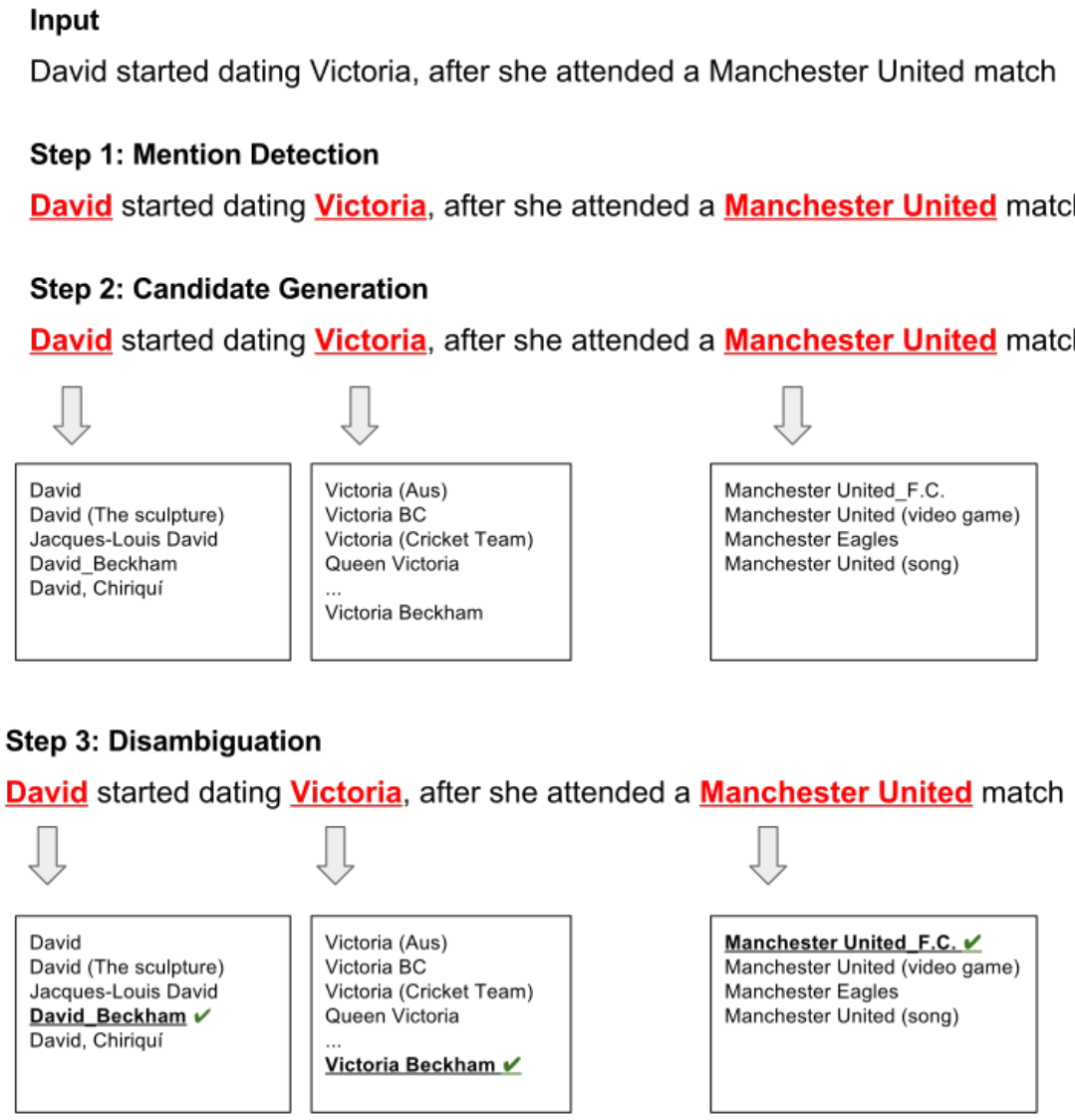


Figure 4.2: Different steps of entity linking for the example: "*David started dating Victoria after she attended a Manchester United match*"

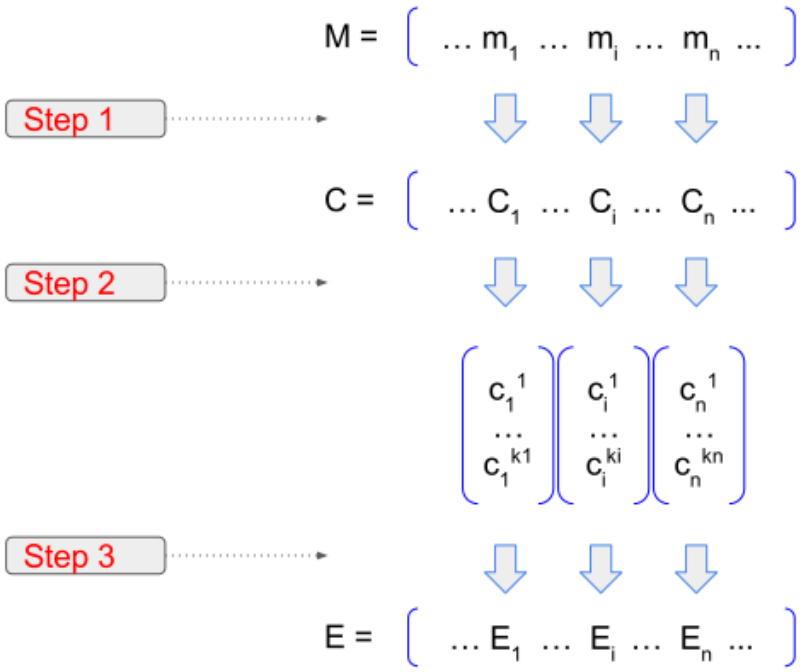


Figure 4.3: A *mention* (m_i) is an ambiguous string that potentially refers to any *candidate* (c_i^j), but only one sense is the correct sense (E_i) in a given sentence.

Finding an optimal solution for Eq. 4.1 is NP-complete and we proceed by transforming it to an Integer Linear Program, following [176, 100, 175].

We start by transforming Eq. 4.1 into the equivalent form of

$$\mathcal{E}^* = \arg\max \sum_{i<j} s_{i,j}^{k,l} r_{i,j}^{k,l} \tag{4.2}$$

where the binary variable $s_{i,j}^{k,l} = 1$ if and only if both c_i^k and c_j^l are selected ($c_i^k \in \mathcal{E}^*$ and $c_j^l \in \mathcal{E}^*$), and $r_{i,j}^{k,l}$ denotes the similarity between the selected pair of entities:

$$r_{i,j}^{k,l} = r(c_i^k, c_j^l) \tag{4.3}$$

The next step is to ensure that every mention e_i is resolved to one and only one candidate c_i^k. This is done by introducing a new binary variable $s_i^k \in \{0, 1\}$ to denote whether or not e_i is resolved to c_i^k. We first need to make sure $s_{i,j}^{k,l} = 1$ will induce both s_i^k and s_j^l are also set to 1:

$$\begin{aligned} \forall i, j, k, l : & s_{i,j}^{k,l} < s_i^k \text{ and} \\ & s_{i,j}^{k,l} < s_j^l \end{aligned} \tag{4.4}$$

Finally, the following constraint will make sure that one and only one c_i^k is assigned to e_i:

$$\forall i \sum_k s_i^k = 1 \tag{4.5}$$

This gives the following IP:

$$\begin{aligned}
&\text{Maximize } \sum_{i<j} s_{i,j}^{k,l} r_{i,j}^{k,l} \text{ for } 1 \leq i,j \leq n, 1 \leq k \leq k_i, 1 \leq l \leq k_j \\
&\text{subject to: } s_i^k \in \{0,1\} \\
&\qquad\qquad s_{i,j}^{k,l} \in \{0,1\} \\
&\qquad\qquad \forall i \sum_k s_i^k = 1 \\
&\qquad\qquad \forall i,j,k,l : s_{i,j}^{k,l} < s_i^k \text{ and } s_{i,j}^{k,l} < s_j^l
\end{aligned} \tag{4.6}$$

4.5 Key Entity Modelling

Lazic et al. [103] conjecture that, for each mention, there is usually one context word that suffices to disambiguate it. Using this assumption, they provide a probabilistic model, referred to as *selective context model*. Motivated by this, we advance the idea and conjecture that, given a short sentence, there exists one entity, referred to as *key entity* in this thesis, that can help disambiguate every other one. With the *key entity* denoted by e^*, this assumption will lead to a different formulation for *coherence* (referred to as *key coherence* in this text):

$$(e^*, \mathcal{E}^*) = \underset{\substack{\epsilon \in \mathcal{C} \\ \hat{\mathcal{E}} \in (\mathcal{C}_1 \times \cdots \times \mathcal{C}_n)}}{\arg\max} \sum_{i=1}^{n} r(\epsilon, \hat{e}_i) \tag{4.7}$$

Unlike Eq. 4.1, there is a quadratic-time solution for this optimization. We first assign to each entity $\epsilon_i \in \mathcal{C}$ a best candidate list $\mathcal{B}_i = [b_i^1, ..., b_i^n]$ where b_i^j is the best candidate for m_j assuming ϵ_i be the *key entity*, i.e, the most similar entity to ϵ_i in $\mathcal{C}_j$. Next, we find the list $\mathcal{B}_i$ with maximum coherence w.r.t its corresponding key, ϵ_i:

$$\begin{aligned}
&\text{Step 1: } \forall \epsilon_i \in \mathcal{C}, \mathcal{B}_i = \left[\underset{t \in \mathcal{C}_j}{\arg\max}\, r(\epsilon_i, t) \mid \forall j \leq n \right] \\
&\text{Step 2: } \mathcal{E}^* = \underset{\mathcal{B}_i}{\arg\max} \sum_{t \in \mathcal{B}_i} r(\epsilon_i, t)
\end{aligned} \tag{4.8}$$

Both steps have $O(|\mathcal{C}|^2)$ complexity. Our experiments demonstrate the effectiveness of this method, thus add support to the conjecture that every sentence has a *key entity*.

4.6 VSM-Based Context-Vector Method

While solving Eq. 4.7 is computationally faster, the improvement over IP based model is not dramatic. Key entity model, similar to IP, needs to calculate $|\mathcal{C}|^2$ similarities, which can be very expensive even with fast relatedness methods. Both our embedding and WLM have $n \log n$ complexity where n is the number of neighbours (and assuming that the embeddings are computed offline). Moreover, the large size of Wikipedia and hardware limitations lead to I/O delay in accessing those neighbours, making the whole process expensive. However, the insight that this approach gives is promising and motivates us to develop a fast *key entity* based method for disambiguation using vector space model (VSM) operations. We start from a simple vector space model-based method to demonstrate both the quality of the vectors, and more importantly, how using vector space embeddings can help finding a faster algorithm.

The first VSM-based approach is a basic "Lesk Like" method: to disambiguate a mention m, the context in which m occurs is compared to the profiles of the candidates, and the one with the highest similarity is chosen. Unlike classic cases of context vector [143], we do not have any representation for words, including mentions; we originally defined the representation $R(\cdot)$ for entities only. However, this notion can easily be extended to mentions:

Definition 4.1 *A mention representation of m, $\mathcal{R}(m)$, is defined to be the average of its candidate representations:*

$$\mathcal{R}(m_i) = \frac{1}{|\mathcal{C}_i|} \sum_{t \in \mathcal{C}_i} \mathcal{R}(t) \tag{4.9}$$

Having defined this notion, defining a context vector is straightforward. The *context vector of a mention* should encode the information about the other mentions in the context. We suggest the following definition for a *context vector* w.r.t to a mention:

Definition 4.2 *The context Vector w.r.t. mention m_i, $\hat{\mathcal{R}}(m_i)$, is the average of other mention representations in the sentence:*

$$\hat{\mathcal{R}}(m_i) = \frac{1}{n-1} \sum_{m_j \in \mathcal{M} \setminus m_i} \mathcal{R}(m_i) \tag{4.10}$$

At this point we can define the *context-vector disambiguation* in three steps:

1. Calculate *mention representations* $\mathcal{R}(m_i)$ for all $i = 1 \dots n$

2. Extract *context vectors* $\hat{\mathcal{R}}(m_i)$ for all $i = 1 \dots n$

3. The disambiguated entities $\mathcal{E}^*$, are those with maximum cosine similarity to their context:

$$\mathcal{E}^* = \left[\arg\max_{t \in \mathcal{C}_i} \mathcal{R}(t) \cdot \hat{\mathcal{R}}(m_i) \mid i = 1 \dots n \right] \tag{4.11}$$

4.7 VSM Key Entity (Key-Coherence) Recognition

Context vector can be roughly considered as the vector space equivalent to the *standard coherence* method. In this section, using this basic method and inspired by the existence of a *key entity*, we aim to develop a VSM-based *key entity* method that tries to guess the *key entity* by benefiting from the vector representations. The idea is to assume the *key entity* to be the one that can be resolved with the highest *certainty*.

Having defined a context vector for every mention, we can sort the candidates for each mention based on their similarity to the correspondent context vector. This value will reflect the degree of relevance of a candidate. Now let's assume that for each mention m_i, k_i^j is the *j-th best candidate* w.r.t. its context vector:

$$k_i^j = \arg\max_{t \in \mathcal{C}_i \setminus \{k_i^1, \cdots, k_i^{j-1}\}} \mathcal{R}(t) \cdot \hat{\mathcal{R}}(m_i) \tag{4.12}$$

We assume that the difference between the similarities of k_i^1 and k_i^2 to the context representation of m_i, $\hat{\mathcal{R}}(m_i)$, is a good indicator of how *certain* we are in resolving m_i to the entity k_i^1. This concept is illustrated in Fig 4.4.

Definition 4.3 *The confidence value for the best candidate for mention m_i, $conf(k_i^1)$, is defined to be the proportional difference between its similarity to the context vector and the similarity of the second best candidate to the context vector:*

$$conf(k_i^1) = \frac{\mathcal{R}(k_i^1) \cdot \hat{\mathcal{R}}(m_i) - \mathcal{R}(k_i^2) \cdot \hat{\mathcal{R}}(m_i)}{\mathcal{R}(k_i^2) \cdot \hat{\mathcal{R}}(m_i)} \tag{4.13}$$

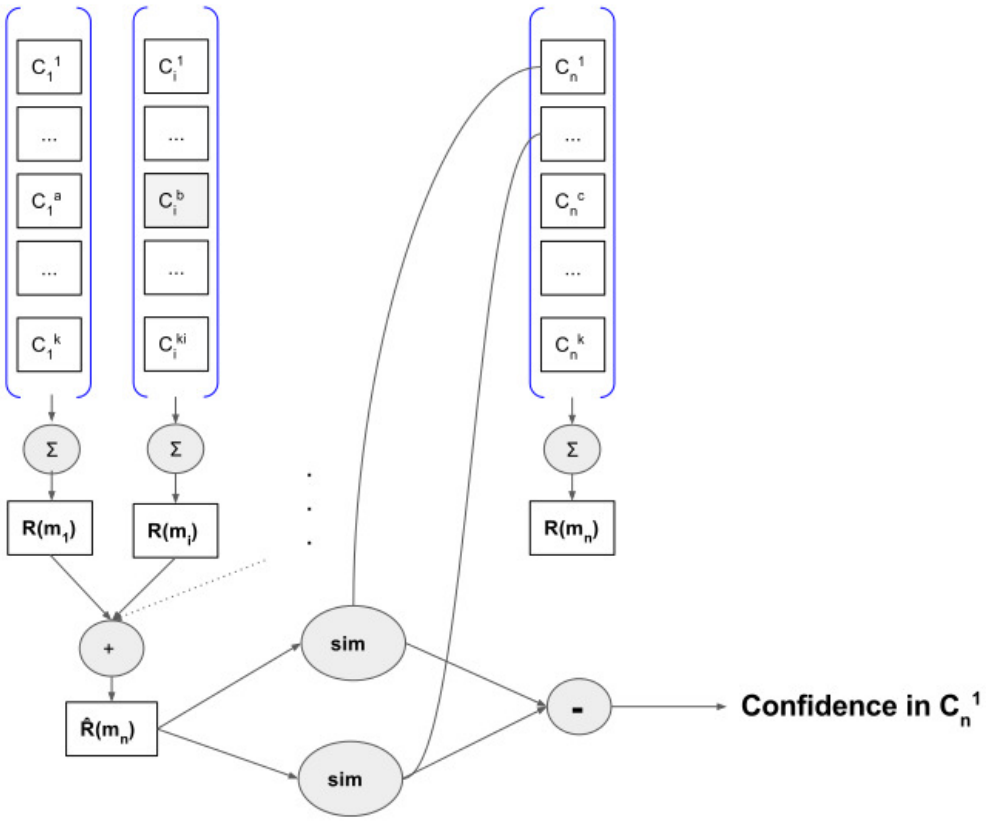


Figure 4.4: Calculating *confidence* for each *best* candidate

Having defined the *confidence* value, we can summarize the VSM *key entity* disambiguation algorithm in five steps.

1. Extract *mention representations*, $\mathcal{R}(m_i)$, for all $i = 1 \dots n$
2. Extract *context vectors*, $\hat{\mathcal{R}}(m_i)$, for all $i = 1 \dots n$
3. Calculate *confidence*, $conf\,(k_i^1)$, for all $i = 1 \dots n$
4. Find *key entity*: We define the *key entity* e^* to be the one that has the highest confidence value:
$$e^* = k_i^1 \text{ where } i = \arg\max_{i \leq n} \left\{ conf\,(k_i^1) \right\} \tag{4.14}$$
5. Disambiguate: Once e^* is found, the disambiguated entities $\mathcal{E}^*$, are those with maximum $cosine$ similarity to $\mathcal{R}(e^*)$:
$$\mathcal{E}^* = \left[\arg\max_{t \in \mathcal{C}_i} \mathcal{R}(t) \cdot \mathcal{R}(e^*) \mid i = 1 \dots n \right] \tag{4.15}$$

Our VSM-based *key entity* recognition method has a linear complexity in terms of the number of candidates $|\mathcal{C}|$ and also calculates only $2|\mathcal{C}|$ similarities. Surprisingly, the experiments demonstrate that this method, despite its sub-optimality in terms of coherence, can outperform previous models in terms of precision. This is not a contradiction: coherence is measured to predict precision but does not imply it.

4.8 A Walk-Through Example

Let's consider the example we started with, "*David started dating Victoria after she attended a Manchester United match*". This sentence has three mentions, *David*, *Victoria* and *Manchester United*. We assume each mention has up to 4 candidates (to save space), as illustrated in Table 4.1.

4.8.1 Coherence Optimization using Integer Programming

The first method we evaluate is the *full coherence model*, in which we search for three entities for which the sum of pairwise similarities is maximum. In Table 4.2, we calculated the pairwise similarities between different candidates. Choosing two candidates for the

David	**Victoria**	**Manchester United**
$\boldsymbol{C_0}$	$\boldsymbol{C_1}$	$\boldsymbol{C_2}$
c_0^0 David	c_1^0 Victoria_(Australia)	c_2^0 Manchester_United_F.C.
c_0^1 David_(Michelangelo)	c_1^1 Victoria,_ British_Columbia	c_2^1 Manchester_United_F.C_ _Reserves_and_Academy
c_0^2 Jacques-Louis_David	c_1^2 Victoria_cricket_team	c_2^2 Manchester_United_ (video_game_series)
c_0^3 David_Beckham	c_1^3 Victoria_Beckham	c_2^3 Manchester_Eagles

Table 4.1: Three mentions, and four candidate per mention for the example sentence "*David started dating Victoria after she attended a Manchester United match*"

same mention is not possible (and is coded in one of the IP constraints). Therefore, the similarity between them is not calculated (marked with a '-' in the table).

	$\boldsymbol{c_0^0}$	$\boldsymbol{c_0^1}$	$\boldsymbol{c_0^2}$	$\boldsymbol{c_0^3}$	$\boldsymbol{c_1^0}$	$\boldsymbol{c_1^1}$	$\boldsymbol{c_1^2}$	$\boldsymbol{c_1^3}$	$\boldsymbol{c_2^0}$	$\boldsymbol{c_2^1}$	$\boldsymbol{c_2^2}$	$\boldsymbol{c_2^3}$
$\boldsymbol{c_0^0}$	-	-	-	-	0.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0
$\boldsymbol{c_0^1}$	-	-	-	-	1.0	1.0	0.0	1.0	1.0	1.0	0.0	0.0
$\boldsymbol{c_0^2}$	-	-	-	-	1.0	1.0	0.0	1.0	4.0	3.0	0.0	0.0
$\boldsymbol{c_0^3}$	-	-	-	-	2.0	9.0	0.0	850.0	1415.0	840.0	7.0	12.0
$\boldsymbol{c_1^0}$	0.0	1.0	1.0	2.0	-	-	-	-	4.0	5.0	2.0	0.0
$\boldsymbol{c_1^1}$	1.0	1.0	1.0	9.0	-	-	-	-	6.0	4.0	1.0	13.0
$\boldsymbol{c_1^2}$	0.0	0.0	0.0	0.0	-	-	-	-	2.0	0.0	1.0	0.0
$\boldsymbol{c_1^3}$	1.0	1.0	1.0	850.0	-	-	-	-	19.0	13.0	0.0	0.0
$\boldsymbol{c_2^0}$	1.0	1.0	4.0	1415.0	4.0	6.0	2.0	19.0	-	-	-	-
$\boldsymbol{c_2^1}$	0.0	1.0	3.0	840.0	5.0	4.0	0.0	13.0	-	-	-	-
$\boldsymbol{c_2^2}$	0.0	0.0	0.0	7.0	2.0	1.0	1.0	0.0	-	-	-	-
$\boldsymbol{c_2^3}$	0.0	0.0	0.0	12.0	0.0	13.0	0.0	0.0	-	-	-	-

Table 4.2: Pairwise similarities between all candidates ($\times 10^{-4}$). The similarities between candidates of the same mention are not calculated (marked with a '-'), as they are not used. We used the symmetry of the table and calculated the upper triangular submatrix only.

Feeding Table 4.2 to the IP given in Eq. 4.6 and solving it yields an optimal value of 0.2284 , which is the result of the following summation:

$$\begin{aligned} optimum &= r(c_0^3, c_1^3) + r(c_0^3, c_2^0) + r(c_1^3, c_2^0) \\ &= (850.0 + 1415.0 + 19.0) \times 10^{-4} \\ &= 0.2284 \end{aligned} \tag{4.16}$$

This leads to the following solution:

$$\begin{aligned} Answer &= [c_0^3, c_1^3, c_2^0] \\ &= [David\_Beckham, Victoria\_Beckham, Manchester\_United\_F.C.] \end{aligned} \tag{4.17}$$

4.8.2 Key Entity Based Disambiguation

In this method, we need to calculate the *key coherence* (Eq. 4.7) assuming any entity is (hypothetically) *the key entity*. For example, if the *key entity* is c_2^0 (*Victoria_(Australia)*), then by referring to Table 4.2, among the candidates for *David*, c_0^3 has the maximum relatedness with the *key entity*, and among the candidates for Manchester_United, c_2^1 has the highest relatedness to the *key entity*. This results in the following *key coherence*:

$$\begin{aligned} key\text{-}coherence(c_2^0) &= r(c_1^0, c_0^3) + r(c_1^0, c_2^1) \\ &= (2.0 + 5.0) \times 10^{-4} \\ &= 0.007 \end{aligned} \tag{4.18}$$

We have calculated these values (the result of disambiguation and the coherence) for each possible candidate, shown in Table 4.3. As we can see, the maximum *coherence* belongs to c_0^3 (*David Beckham*), which leads to resolving Victoria to c_1^3 (Victoria Beckham) and *Manchester United* to c_2^0 (*Manchester United F.C.*).

4.8.3 VSM-Based Methods: Context-Vector and Key Entity Based

The last two methods that we analyze are the vector-space based methods, in which we abandon the *entity-entity* Table 4.2 with the size of $n^2 = 144$ in favour of an *entity-context* Table with a size of $n = 12$. Each entry of this table contains two values, a candidate and the relatedness between this candidate and its context vector. For example, the value that

		David	Victoria	Manchester United
C_{0j}	**Key**	c_0^0	c_1^0	c_2^0
	Resolved Mentions	$[c_0^0,c_1^1,c_2^0]$	$[c_0^3,c_1^0,c_2^1]$	$[c_0^3,c_1^3,c_2^0]$
	Key-Coherence Value	0.0003	0.0007	0.1434
C_{1j}	**Key**	c_0^1	c_1^1	c_2^1
	Resolved Mentions	$[c_0^1,c_1^0,c_2^0]$	$[c_0^3,c_1^1,c_2^3]$	$[c_0^3,c_1^3,c_2^1]$
	Key-Coherence value	0.0002	0.0022	0.0854
C_{2j}	**Key**	c_0^2	c_1^2	c_2^2
	Resolved Mentions	$[c_0^2,c_1^3,c_2^0]$	$[c_0^3,c_1^2,c_2^0]$	$[c_0^3,c_1^0,c_2^2]$
	Key-Coherence Value	0.0005	0.0002	0.0009
C_{3j}	**Key**	c_0^3	c_1^3	c_2^3
	Resolved Mentions	$[c_0^3,c_1^3,c_2^0]$	$[c_0^3,c_1^3,c_2^0]$	$[c_0^3,c_1^1,c_2^3]$
	Key-Coherence Value	0.2265	0.0869	0.0025

Table 4.3: *Key entity* disambiguation result and key coherence for each entity. Every entry contains a candidate, along with the *key coherence* and the result of disambiguation, assuming that candidate is the *key entity*.

is associated with c_0^3 equals to $\mathcal{R}(c_0^3) \cdot \hat{\mathcal{R}}(c_0^3)$. The candidates are sorted by their *context similarity* value, therefore, this table alone can be used to disambiguate, which is what we explained as *context method*. The top entities in each column are $[c_0^3, c_1^3, c_2^0]$, which are the correct candidates.

	David	Victoria	Manchester United
Key	c_0^3	c_1^3	c_2^0
Context-Similarity	0.0353	0.0075	0.0384
Key	c_0^2	c_1^1	c_2^1
Context-Similarity	0.0001	0.0011	0.023
Key	c_0^1	c_1^0	c_2^3
Context-Similarity	0.0	0.0003	0.0007
Key	c_0^0	c_1^2	c_2^2
Context-Similarity	0.0	0.0001	0.0003

Table 4.4: Entity to context similarity, sorted by decreasing similarity. Each entry contains a candidate and the similarity between it and its correspondent context vector.

To proceed with our VSM based *key entity* method, we need to calculate the confidence for each entity. The confidence value of c_0^3, as an example, is the proportional difference

between its *context similarity* and the *context similarity* of the entity following it, that is:

$$conf(c_0^3) = \frac{\mathcal{R}(c_0^3) \cdot \hat{\mathcal{R}}(c_0^3) - \mathcal{R}(c_0^2) \cdot \hat{\mathcal{R}}(c_0^2)}{\mathcal{R}(c_0^2) \cdot \hat{\mathcal{R}}(c_0^2)} \approx \frac{0.0353 - 0.0001}{0.0001} \approx 360.93 \tag{4.19}$$

Calculated confidence values for all the three top entities are reported in Table 4.5. We can see that the confidence value for c_0^3 is higher than the other two, so c_0^3 is chosen as the key entity. This is the same *key entity* we found using direct search in Section 4.8.2, and therefore, choosing it as the *key entity* similarly leads to resolving Victoria to c_0^3 (Victoria Beckham) and *Manchester United* to c_2^0 (*Manchester United F.C.*).

	David	**Victoria**	**Manchester United**
Key	c_0^3	c_1^3	c_2^0
Confidence	360.93	5.97	0.67

Table 4.5: Confidence value for the top candidates of Table 4.4

4.9 Evaluations

We evaluated our proposed method as well as *graph overlap*, WLM and $word2vec_2$ (Section 3.8.1) with IP on five different datasets:

1. AQUAINT [138]: 50 documents from news. AQUAINT is a corpus of news documents, taken from three news agencies: the New York Times, the Associated Press, and the Xinhua News Agency. A subset, consisting of short documents (250 to 300 words) of the news belonging to New York Times and annotated by humans, was used in the WSD evaluations.

2. MSNBC [44]: 20 news documents. 10 MSNBC news categories were selected (Business, U.S. Politics, Entertainment, Health, Sports, Tech & Science, Travel, TV News, U.S. News, and World New), and from each category, 2 top stories were annotated by humans.

3. Kore [78]: 50 human-curated hard sentences. This dataset is hand-crafted and meant to be more difficult than the other datasets. They have short context and higher

density of mentions (3 mentions per sentence, and a mention to word ratio of 20%), are highly ambiguous (on average more than 600 candidates per mention), and most importantly, includes more mentions whose correct targets are less popular, unlike the news documents where most of the time the correct entity is the most popular one.

4. CoNLL annotated by the authors of [80]: 1393 articles. Only the proper nouns from Reuters newswire articles are hand annotated and linked to YOGA2 [79] (hence sometimes called CoNLL-YAGO to avoid confusion with the original CoNLL dataset).

5. Wiki.5000 (or simply Wiki): The first 5000 articles (ordered by their *id*) of Wikipedia (Wiki dataset). Because entities are not necessarily linked throughout the whole article, we only chose the opening paragraphs, which tend to have a higher quality[1]. We made sure that at least 10% of the documents contain a mention whose target entity is not the most popular one.

We experiment with candidate lists of size 5, 10 and 15 (any number beyond 15 was not feasible with our hardware); the list always contains the correct entity. Sentences are chunked and each chunk contains up to 5 mentions.

Micro Average Precision ($\hat{\pi}^{\mu}$) and Macro average Precisions ($\hat{\pi}^{M}$) are reported over the five datasets and different candidate numbers, defined as follows [185]:

$$\hat{\pi}^{\mu} = \frac{\sum_{i=1}^{d} TP_i}{\sum_{i=1}^{d}(TP_i + FP_i)}$$
$$\hat{\pi}^{M} = \frac{\sum_{i=1}^{d} \hat{\pi}_i}{d} \tag{4.20}$$

where TP_i denotes the number of *true positives*, FP_i the number of *false positives*, $\hat{\pi}_i$ is the *precision* w.r.t. the *i*th document ($\hat{\pi}_i = \frac{TP_i}{TP_i+FP_i}$) and d is the size of the collection.

4.9.1 Standard Coherence: Evaluation of Our Relatedness Method

We plugged different relatedness methods into IP (Eq. 4.6) and solved the equation to find the entities. The results are shown in Table 4.6. Our embedding-based method outperforms other methods in most of the datasets and this trend stays the same with the increase of the number of candidates.

[1]The dataset is publicly available on the project website.

As mentioned before, we always made sure that the correct entity is in the list (if not, we just added it), and that explains the decrease in the results when the number of candidates increases: the more the number of the candidates, the harder is to find the correct entity. In this section, we are only interested in the quality of the relatedness and that justifies our decision. However, in the next chapter, we explain a Wikifier which describes a real-word system using the modules developed here.

		MSNBC		AQUAINT		KORE		CONLL		WIKI	
# Cands	Sim-Method	$\hat{\pi}^{\mu}$	$\hat{\pi}^{M}$	$\hat{\pi}^{\mu}$	$\hat{\pi}^{M}$	$\hat{\pi}^{\mu}$	$\hat{\pi}^{M}$	$\hat{\pi}^{\mu}$	$\hat{\pi}^{M}$	$\hat{\pi}^{\mu}$	$\hat{\pi}^{M}$
5	overlap	.83	.82	.64	.64	.75	.74	.70	.68	.74	.66
	wlm	.85	.83	.69	.69	.80	.78	.73	.67	.77	.68
	$word2vec_2$	.85	.84	.63	.62	.83	.81	.70	.69	.76	.69
	rvsPageRank	**.87**	**.86**	**.71**	**.72**	**.86**	**.86**	**.73**	**.73**	**.80**	**.72**
10	overlap	.77	.78	.55	.55	.69	.68	.62	.59	.67	.60
	wlm	.79	.78	.58	.59	.77	**.76**	.64	.58	.71	.62
	$word2vec_2$	.79	.79	.54	.53	.77	.74	.65	.63	.71	.64
	rvsPageRank	**.82**	**.82**	**.64**	**.64**	**.77**	.75	**.67**	**.66**	**.76**	**.67**
15	overlap	.75	.76	.50	.51	.62	.62	.57	.52	.64	.57
	wlm	.76	.76	.54	.55	.73	**.72**	.60	.54	.68	.59
	$word2vec_2$	.77	.77	.49	.49	.67	.65	.61	.57	.68	.61
	rvsPageRank	**.80**	**.80**	**.59**	**.59**	**.75**	**.72**	**.63**	**.61**	**.73**	**.65**

Table 4.6: WSD using Integer Programming (IP): Micro Averaged Precision ($\hat{\pi}^{\mu}$) and Macro Averaged Precision ($\hat{\pi}^{M}$), using different semantic relatedness methods, across different candidate numbers and datasets

4.9.2 The Proposed Key Entity Method vs Standard Coherence Model

We report the comparison of the results of the *key coherence* model with the coherence model (solved using IP) in Table 4.7. Both methods are competitive, which lends support the idea that a *key entity* exists. Especially on the last two datasets, where the sizes are significantly larger, one can notice that the *key entity* method is as good as or better than the standard dataset. We will see in the next section that we can improve on these results using our VSM *key entity* model.

4.9.3 Using The Vector Space Model To Disambiguate

In this section, we apply our VSM models to evaluate the quality of the vectors and the effectiveness of the *key entity* method. As we mentioned before, we can roughly assume

# Cands	Sim-Method	MSNBC $\hat{\pi}^{\mu}$	MSNBC $\hat{\pi}^{M}$	AQUAINT $\hat{\pi}^{\mu}$	AQUAINT $\hat{\pi}^{M}$	KORE $\hat{\pi}^{\mu}$	KORE $\hat{\pi}^{M}$	CONLL $\hat{\pi}^{\mu}$	CONLL $\hat{\pi}^{M}$	WIKI $\hat{\pi}^{\mu}$	WIKI $\hat{\pi}^{M}$
5	Coherence*	.87	.86	.71	.72	.86	.86	.73	.73	.80	.72
	Key-Coherence	.87	.85	.70	.70	.87	.86	.73	.74	.81	.81
10	Coherence*	.82	.82	.64	.64	.77	.75	.67	.66	.76	.67
	Key-Coherence	.82	.81	.61	.61	.79	.77	.67	.67	.76	.76
15	Coherence*	.80	.80	.59	.59	.75	.72	.63	.61	.73	.65
	Key-Coherence	.80	.79	.56	.56	.73	.73	.62	.62	.74	.74

Table 4.7: Comparing the results of Integer Programming (IP) with *key entity* based method: Micro Averaged Precision ($\hat{\pi}^{\mu}$) and Macro Averaged Precision ($\hat{\pi}^{M}$), across different candidate numbers and datasets. *: extracted from Table 4.6)

that the *context based* method (cf. section 4.6) is the VSM equivalent of the *coherence* method and the *VSM key entity* method (cf. Section 4.7) is the VSM equivalent of *key entity* method. Therefore, we include the results from the previous tables (Table 4.6 and Table 4.7) to be able to compare them side by side. The results in Table 4.8 demonstrate that the vector space model can be very competitive, and often outperforms equivalent semantic relatedness-based methods. One can also notice that in the majority of cases, especially when the number of candidates is at its maximum (15), the VSM key entity-based method achieves the best results. The main strength of the model is its speed, reported in Fig. 4.5 (log scale). A better performance with a dramatic speedup (more than 50x) provides more evidence for the quality of the embeddings.

4.9.4 Evaluating the Quality of *word2vec* Embeddings in the VSM Based Methods

Both the context similarity method and our introduced VSM *key entity* method can be used with any other embedding and in fact can provide yet another method for evaluating the quality of our vectors. We re-evaluate $word2vec_2$ vectors again, this time using the VSM based method. The results in Table 4.9 demonstrate that our embeddings can outperform the *word2vec* embeddings, with an even higher margin than in the previous IP-based experiments. We believe that the under performance of $word2vec$ in this approach is caused by the density of the vectors; dense vectors are more susceptible to noise when averaged than sparse vectors because they have a lower capacity to contain information. This is more

# Cands	Sim-Method	MSNBC $\hat{\pi}^{\mu}$	MSNBC $\hat{\pi}^{M}$	AQUAINT $\hat{\pi}^{\mu}$	AQUAINT $\hat{\pi}^{M}$	KORE $\hat{\pi}^{\mu}$	KORE $\hat{\pi}^{M}$	CONLL $\hat{\pi}^{\mu}$	CONLL $\hat{\pi}^{M}$	WIKI $\hat{\pi}^{\mu}$	WIKI $\hat{\pi}^{M}$
5	Coherence*	**.87**	**.86**	.71	.72	.86	.86	.73	.73	.80	.72
	Context	.77	.74	**.73**	**.74**	.87	.88	.62	.61	**.82**	.81
	Key Entity†	**.87**	.85	.70	.70	.87	.86	.73	.74	.81	.81
	VSM Key Entity	**.87**	.85	.70	.71	**.89**	**.90**	**.81**	**.79**	**.82**	**.82**
10	Coherence*	**.82**	**.82**	.64	.64	.77	.75	.67	.66	.76	.67
	Context	.74	.74	**.67**	**.67**	.78	**.77**	.58	.56	.77	.77
	Key Entity†	**.82**	.81	.61	.61	**.79**	**.77**	.67	.67	.76	.76
	VSM Key Entity	**.82**	**.82**	.64	.64	.77	.76	**.76**	**.73**	**.78**	**.78**
15	Coherence*	.80	.80	.59	.59	.75	.72	.63	.61	.73	.65
	Context	.71	.69	**.62**	**.63**	.73	.73	.55	.54	.74	.74
	Key Entity†	.80	.79	.56	.56	.73	.73	.62	.62	.74	.74
	VSM Key Entity	**.81**	**.81**	.59	.59	**.75**	**.74**	**.73**	**.70**	**.76**	**.75**

Table 4.8: Comparing the results of Integer Programming (IP) with its equivalent VSM model (*context similarity*), and *key entity*-based method with its equivalent VSM model (VSM key entity): Micro Averaged Precision ($\hat{\pi}^{\mu}$) and Macro Averaged Precision ($\hat{\pi}^{M}$), across different candidate numbers and datasets. *: extracted from Table 4.6, †: extracted from Table 4.7

obvious when we notice that when the number of candidates is increased to 15, *word2vec* is the inferior method in all of the cases.

4.10 Conclusion

We evaluated our concept representations in a word sense disambiguation task by using a standard coherence modeled based on Integer Programming (IP) and demonstrated that it performs better than the other methods used in this task. Moreover, by reformulating coherence, we demonstrated that there is often one *key entity* in a sentence that can help with disambiguating the rest of the entities. This finding led to a very fast yet more accurate method for WSD that we refer to as *VSM key entity recognition*. We make available the concept embeddings for public use so that they can easily be incorporated in any NLP relatedness task with minimum overhead.

			MSNBC		AQUAINT		KORE		CONLL		WIKI	
#⋆	Sim-Method		$\hat{\pi}^{\mu}$	$\hat{\pi}^{M}$	$\hat{\pi}^{\mu}$	$\hat{\pi}^{M}$	$\hat{\pi}^{\mu}$	$\hat{\pi}^{M}$	$\hat{\pi}^{\mu}$	$\hat{\pi}^{M}$	$\hat{\pi}^{\mu}$	$\hat{\pi}^{M}$
5	Context	word2vec	**.78**	**.76**	.61	.60	.79	.78	**.66**	**.64**	.75	.76
		rvsPageRank†	.77	.74	**.73**	**.74**	**.87**	**.88**	.62	.61	**.82**	**.81**
	VSM Key	word2vec	.80	.80	.59	.59	.82	.80	.68	.66	.76	.76
		rvsPageRank†	**.87**	**.85**	**.70**	**.71**	**.89**	**.90**	**.81**	**.79**	**.82**	**.82**
10	Context	word2vec	.69	.68	.50	.49	.55	.57	**.58**	**.56**	.68	.69
		rvsPageRank†	**.74**	**.74**	**.67**	**.67**	**.78**	**.77**	**.58**	**.56**	**.77**	**.77**
	VSM Key	word2vec	.75	.76	.52	.51	.63	.62	.62	.59	.71	.71
		rvsPageRank†	**.82**	**.82**	**.64**	**.64**	**.77**	**.76**	**.76**	**.73**	**.78**	**.78**
15	Context	word2vec	.65	.63	.46	.45	.44	.46	.54	.51	.64	.65
		rvsPageRank†	**.71**	**.69**	**.62**	**.63**	**.73**	**.73**	**.55**	**.54**	**.74**	**.74**
	VSM Key	word2vec	.72	.75	.49	.49	.54	.54	.58	.55	.68	.69
		rvsPageRank	**.81**	**.81**	**.59**	**.59**	**.75**	**.74**	**.73**	**.70**	**.76**	**.75**

Table 4.9: Comparing the quality of the vectors of our *rvsPagerank* embedding with *word2vec*. Both of the embeddings are evaluated in two VSM-based methods, Context Similarity and VSM Key Entity: Micro Averaged Precision ($\hat{\pi}^{\mu}$) and Macro Averaged Precision ($\hat{\pi}^{M}$), across different candidate numbers and datasets. †: extracted from Table 4.8

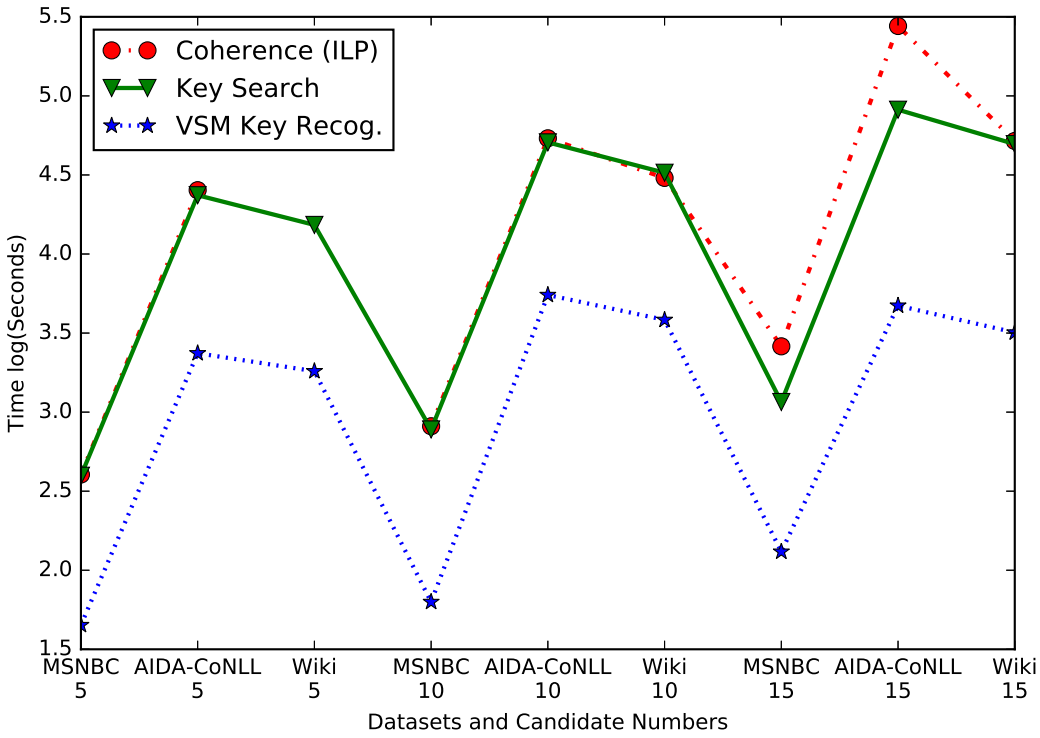


Figure 4.5: linear-log plot of time spent for the three largest datasets

Chapter 5

Wikification

5.1 Introduction

A large number of Natural Language Processing (NLP) systems rely on Lexical Knowledge Resources (LKR). This includes simple tasks such as exploring synonyms of the terms in a given text using WordNet [136], to more sophisticated analyses using UMLS ontologies [21]. Wikipedia is gaining a lot of attention in NLP and recent studies have shown that it is comparable to domain-specific ontologies in some specific tasks [181].

Using any LKR with unstructured text requires a customized *entity linker* to detect *entities* (or concepts) in the text and relate them to the knowledge source. In the case of UMLS for example, a widely used tool called MetaMap [12] is provided as part of the framework. When the LKR is Wikipedia, the entity linker is also referred to as a *Wikifier*. Some of the successful examples are Wikipedia Miner [138], GLOW [168] and tagME [46, 160].

One standard and widely used approach in most of the Wikifiers is to combine several features from the given text and cast it as a classification problem. Most of these features are simple and straightforward, such as *popularity* (the probability of a mention referring to a specific entity), or *commonness* (similarity between a string and an entity). On the other hand, some other features can be more complicated, such as *semantic coherence*. The idea of semantic coherence is to choose a set of entities that are semantically related to each other. The standard definition of coherence tends to be computationally expensive and several studies have proposed heuristics to approximate it [46, 168].

In this study, we propose an entity linker relying on the definition of the *key entity coherence* presented in Chapter 4. The main advantage of *key entity* is its low computational complexity and good performance. It relies on a set of linear vector operations to find a central entity in a short text, the *key entity*, that can assist disambiguating the other entities. The *key entity* based disambiguation relies only on the concept representations extracted from a Wikipedia graph and ignores other information. From this perspective, this chapter

is an attempt to combine the graph information of Wikipedia with its text in an NLP task.

The main outcome of this research is Wikisim Wikifier [1], an implementation of an entity linker. Wikisim consists of several modules for entity recognition and disambiguation. The main module of the system is a *learning-to-rank* module that combines several features from the graph and the text of Wikipedia. The final system can compete with commercial and widely used systems.

5.2 Related Work

One way to summarize the literature on entity linking is following the general framework introduced by two inspiring studies [100] and [168]. In this approach, a Wikifier has access to two classes of information, *local* and *global*. Local features include the context around the entity mention, and some data-driven statistics about the mentions and candidates, such as prior probabilities. Global information usually includes only a *semantic coherence* measure, which represents how coherent the entities in the text are. Adding this second component makes the problem very complex and was in fact ignored by the first Wikifiers. Wikification started as a departure from traditional (proper) Named Entity recognition [26]. In this pioneering research, a SVM-based supervised method was used to learn the proper nouns using the similarity context of the entity and several features from the Wikipedia pages (Lesk-like [107]) and their categories; one SVM per entity was trained. Another early work on Wikipedia (and probably the first research to use the term "Wikify") was [131]. They used a similar approach with a few differences, such as using more local features (referred to as "Data Driven Sense Probabilities" in their research), and also Naive Bayes as their learning algorithm.

Coherence was first explored in [44] and later in Wikipedia Miner [138]. Both of these methods tackle the complexity by selecting a reference *disambiguation context*, i.e, a reference set of entities that other entities are chosen to be consistent with, in order to preserve coherence. Some examples of disambiguation context are the set of all candidates [44] or the set of unambiguous entities in Wikipedia Miner [138]. Wikipedia Miner is one of the first, and most widely used Wikifiers, mainly because it provided a robust open-source web service. The relatedness method used in [138] to model coherence is their own WLM [202],

[1] `https://github.com/asajadi/wikisim`

which is by far the most widely used Wikipedia-based relatedness method. Another approach to use coherence is to turn it into a local feature, for example by extending the *concept-concept* relatedness to *mention-concept* relatedness, by defining it as the average of the similarity of all the mention candidates to a given concept [73]. GLOW [168] uses a supervised method trained on local features only to find an initial set of disambiguation contexts and then performing another round of training to find the final ranking. However, GLOW performs a final pruning using another supervised model to decide whether a mention should be linked to NIL.

This general local/global approach can be used in any other entity-linking system. For example, Tulip [114] is an Entity Recognition and Disambiguation (ERD) system that uses Freebase as the target knowledge source. Tulip mainly focuses on popularity and builds the disambiguation context with the set of the most popular senses. Later, individual candidates are compared to this disambiguation context and the most similar ones are picked.

A slightly different wikifying system is AIDA [80]. It still follows the same local and *coherence* features, but the optimization is done in a graph-based fashion. A graph is constructed between mentions and candidates, and the solution is a subgraph that contains all the mentions, and includes only one candidate for each mention. Another difference is in the similarity metric they use, which is their proposed Keyphrase Overlap Relatedness (KORE) [78].

Another widely used and one of the few well-maintained Wikifiers is tagME [46] and its subsequent versions such as TagME2 and WAT [160]. WAT uses SVM to learn to detect mentions. For disambiguation and final pruning, it uses several approaches. Originally, tagME used a voting algorithm where each entity gets votes from other candidates and the one with the highest vote is chosen. The vote is a combination of semantic relatedness and local features of the entity. It also uses the traditional and successful Random Walk on a mention-entity graph [8, 2].

Further improvements are possible by either using more suitable learning algorithms, such as Probabilistic Bag of Hyperlink (PBoH) [57] or neural networks [81, 53], or by incorporating even more resources, such as “search engine piggybacking” [42].

5.2.1 Learning to Rank

"*Learning to Rank*" is a form of supervised learning where the task is to *rank* (denoted by a '$\prec$' notation) a set of given objects. *Learning to rank* has many applications, but it is particularly important in search engines and information retrieval. As an example, what matters regarding the results of a search engine is the order in which the information is presented, not the particular scores. There are three main approaches to this problem: *pointwise*, *pairwise* and *listwise*. These approaches are different in their *input/output spaces*, *hypotheses*, and *loss functions* [115]:

1. *pointwise*. It is the most similar approach to traditional regression, or even more, to *ordinal regression*. The input is the *feature vector* and the output space consists of the relevance scores.

2. *pairwise*. The input is a pair of instances (i_1, i_2) and the output space is $\{0, 1\}$, representing the truth value of $i_1 \prec i_2$.

3. *listwise*. The input is a *set*, and the output is a *ranked list* (with or without the relevancy scores). The loss function compares the differences between two ranked lists.

We use the state-of-the-art *LambdaMart learning to rank* algorithm [28], which belongs to a family of *learning to rank algorithms* presented by Microsoft Research. It is a modified version of LambdaRank, which itself is based on RankNet [27]. LambdaMart is a list-wise approach that aims to minimize the ranking error directly. It uses Multiple Additive Regression Trees (Mart) to optimize the ranker. The loss function is Normalized Discounted Cumulative Gain (NDCG), which is a widely used measure for the quality of a ranking in information retrieval [122]. This function is not continuous, hence not differentiable [115]. However LambdaMart manages to optimize it using the key observation that one does not need the actual cost values for optimization, but only the gradients [28, 29].

5.3 Problem Definition

Extending the definitions of [180], we represent a sentence by a list of terms $\mathcal{T} = [t_1, \ldots, t_l]$. A mention list $\mathcal{M} = [m_1, \ldots, m_n]$ is associated with every sentence, where $m_i = [t_{r_i}, \cdots, t_{s_i}]$

and $s_i < r_j$ for every $i < j$. In other words, mentions can extend over multiple words, but cannot overlap.

Each mention m_i can have k_i potential senses or candidates: $\mathcal{C}_i = \{c_i^1, \ldots, c_i^{k_i}\}$ and $\mathcal{C} = \bigcup_i \mathcal{C}_i$ is the set of all candidates. The goal is to find $\mathcal{E} = [e_1, \ldots, e_n]$ where e_i is the "*correct*" entity to which mention m_i refers.

5.4 Mention Detection

Many successful Wikifiers rely on well-known named entity recognizers, such as AIDA which uses the Stanford NER Tagger [2] [49] and GLOW which uses Illinois Named Entity Tagger [3] [167]. However, mention detection can be quite complex in the case of Wikipedia. Some of the specific challenges with Wikipedia (compared with other LKRs) are:

1. The number of entities is very large, around 14 million.

2. It is not closed, the entities can be mentioned by anything, and it can even change over time.

3. It is not domain-specific, and including all domains (history, logic, math, etc.) leads to a higher amount of ambiguity, up to the point that everything can be a mention. For example, "is a" can mention the "Is-A" relationship, or "I am" can refer to a poem by "John Clare".

4. It contains a lot of noise. There are mentions that contain a word before (as result of human mistake), or even punctuation, for example "The USA," can be a mention.

5. There can be a lot of possible overlapping mentions. To give a rather strange example, lets say the text contains "!!!". A single "!" is a mention of "'*exclamation mark*' or "*factorial*"; "!!" can be "Brilliant move" in chess or "*double factorial*"; and even "!!!"' is an American band (pronounced as *chk-chk-chk*).

Inspired by the Tulip Named Entity Linker [114], we start the mention detection using a Finite State Transducer (FST). This approach takes a dictionary and matches it to a text. We rely on SolrTextTagger to do this initial matching [4]. The result of this step is quite

[2] `https://nlp.stanford.edu/software/CRF-NER.shtml`
[3] `http://cogcomp.org/page/software_view/NETagger`
[4] `https://github.com/OpenSextant/SolrTextTagger`

noisy. Therefore, we perform another two steps to incrementally prune the results. Overall, our mention detection consists of the following three steps:

1. **FST with SolrTextTagger**. We use SolrTextTagger to extract the mentions using a *longest dominant right* heuristics. This way, the longest string is chosen whenever mentions overlap.

2. **Supervised Model**. In this step, we combine several data-driven statistics. Moreover, we acquire a successful named entity recognizer shipped with "Stanford CoreNLP" [5] [123] as an internal module in our system. This is done by using a supervised model to prune the results from the previous step. The model is a binary Gradient Boosted Classifier [55, 74] trained on a corpus consisting of $30,000$ Wikipedia opening texts. The classifier is trained using the following features:

   (a) The POS tags *before*, *on*, and *after* the word. Although we use the combination of all three of them, but it is obvious from Fig. 5.1 that even looking at the POS on the mention can significantly help pruning the less possible ones.

   (b) Mention probability, defined to be $n(m_i = 1)/\sum_{x\in\{0,1\}} n(m_i = x)$, where $n(m_i = 1)$ and $n(m_i = 0)$ denotes the number of times m_i is, or is not a mention, respectively.

   (c) Whether it is picked up by CoreNLP.

   (d) Whether it starts with a capital.

   (e) Whether there is an exact title match in Wikipedia.

   (f) Whether it contains a space.

   (g) Whether it contains only ASCII characters.

3. **Final Pruning**. We realized another round of pruning based on prior probabilities can further improve the results (though this would not be necessary in the presence of sufficient data). In this step, we remove mentions where the most popular candidate has less than 10 links to it.

[5] `https://stanfordnlp.github.io/CoreNLP/`

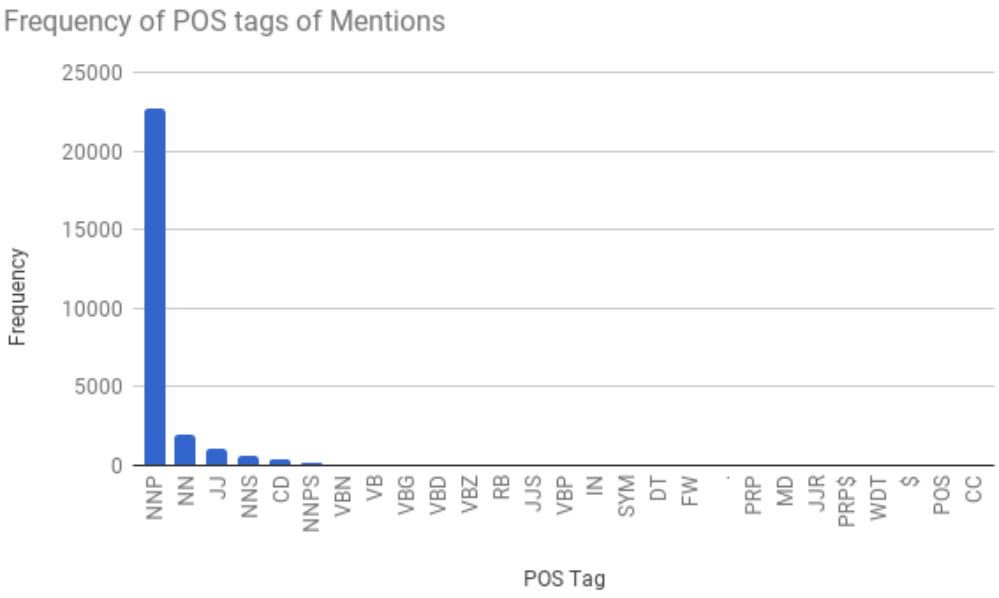


Figure 5.1: The distribution of POS on the mentions

5.5 Disambiguation

For the candidates of each mention, we choose the n most frequently linked entities from that mention. In the experiments we set $n = 20$, where our experiments showed that it includes the correct entity for around 0.85% of the mentions. Disambiguating entities is the final step of entity linking, which is defined as finding the correct target entity for each mention. Following [168], the features we use can be classified as either *local* or *global*. Local features include both contextual information and also statistics regarding the mention-entity relation. Global features usually include semantic coherence. The most important local feature is popularity, that is, the frequency of a given entity being linked to by the given mention. It can influence the results dramatically. However, we need several other features to control its effect in favour of less popular entities. Initially, we have three different features to disambiguate a mention, two distributional (*popularity*, *context relevance*) and one structural (*key entity coherence*).

5.5.1 Popularity

We define this feature to be the probability of a mention being linked to a candidate. For the sake of simplicity, we overload m_i to also be a random variable, so that $m_i = 1$ means that

m_i is actually a mention and $m_i = 0$ to denote that it is not. We can define this probability as:

$$\begin{aligned} p(m_i, c_i^j) &= p(m_i = 0) \times p(c_i^j | m_i = 0) + p(m_i = 1) \times p(c_i^j | m_i = 1) \\ &= \frac{n(m_i = 1)}{\sum_{x \in \{0,1\}} n(m_i = x)} \times \frac{n(c_i^j, m_i)}{\sum_{c \in C_i} n(m_i, c)} \\ &= \frac{n(c_i^j, m_i)}{\sum_{x \in \{0,1\}} n(m_i = x)} \end{aligned} \tag{5.1}$$

where $n(\cdot)$ denotes simple count and $p(m, c)$ and $n(m, c)$ are the probability and the number of times mention m is linked to entity c, respectively. Note that the second term in the $p(c_i^j | m_i = 0) = 0$ and hence, cancels out.

5.5.2 Context Relevance

We tried different approaches to account for context. Two sparse TFIDF approaches and one based on *word2vec*. Given a mention and its context (a fixed window), the idea is to give every entity a relevancy score. We tried three different approaches:

1. **context-article relevance**. This score reflects the similarity between the context and the article associated with the candidate.

2. **context-context relevance**. This score represents the similarity between the mention context and the entity context. For each entity, we compiled a list of contexts, that is, the collection of all windows where this entity is mentioned in Wikipedia.

3. **context-context dense relevance**. This score is the similarity between the average of the word embeddings in the context with the vector assigned to the candidate. We used the embeddings from [188]. In this approach, word2vec is trained on a normalized Wikipedia, i.e, where all mentions are replaced by the target entity.

For the sparse TFIDF similarity, we rely on Lucene [6]. Lucene provides various ways to calculate similarity. However we only used the VSM model. This default similarity is based on TFIDF but accounts for various factors such as the length of the documents and query and also the degree of overlap, i.e, what percentage of the context is covered in the document.

[6] `https://lucene.apache.org`

5.5.3 Coherence

Coherence is the most challenging feature as we explained in the previous section. However, we adopt the *key entity* based coherence definition introduced in the previous chapter and find the key using the VSM based method. Assuming the *key entity* is e^*, the *coherence* of an entity is defined to be simply the similarity to e^*.

5.5.4 Training The Model

Traditionally, a classifier can learn to differentiate between positive and negative labels. The notion of positive and negative label for an entity is only definable when compared with another entity and therefore, learning to rank is more reasonable than a standard classifier; our experiments are consistent with this expectation. We use the *list-wise*, state-of-the-art learning to rank algorithm, LambdaMart [27].

5.6 Experiments

We compared our system with the state-of-the-art entity linker tagME. We only focus on systems with a public interface and tagME is the only currently maintained system to the best of our knowledge. TagME has several generations and we used their provided web service [7] to evaluate it directly. We evaluated our system on three datasets:

1. AQUAINT [138]: 50 news documents.
2. MSNBC [44]: 20 news documents.
3. Kore [78]: 50 human-curated sentences intended to be difficult for entity linking.
4. Wiki5000: The first 5000 articles of Wikipedia (Wiki dataset). Because entities are not necessarily linked throughout the whole article, we only choose the opening paragraphs, which tend to have a higher quality[8].
5. NoPop: 17 records from Kore, and 85 records from Wiki5000, where each record does not contain any mentions where the correct entity is the most popular one. Popularity is a too strong measure for disambiguating entities and can clearly mask the

[7] `https://tagME.d4science.org/tagME/`
[8] The dataset is publicly available on the project website

performance of the system on the not-popular entities, i.e, entities that are not the most probable candidate for a given mention. This inspired us to isolate those entities in the evaluations and NoPop refers to this set of entities.

Also all of the models are trained on the 30000 opening texts of Wikipedia pages. We start from evaluating the individual components of the system, mention detection and disambiguation, and finally we will report the results for the overall system. We report *micro-averaged precision* ($\hat{\pi}^{\mu}$), *micro-averaged recall* ($\hat{\rho}^{\mu}$) and *micro averaged F-measure* (F_1^{μ}): defined as follows [185]

$$\begin{aligned}\hat{\pi}^{\mu} &= \frac{\sum_{i=1}^{d} TP_i}{\sum_{i=1}^{d}(TP_i + FP_i)} \\ \hat{\rho}^{\mu} &= \frac{\sum_{i=1}^{d} TP_i}{\sum_{i=1}^{d}(TP_i + FN_i)} \\ \hat{F}_1^{\mu} &= 2 \cdot \frac{\hat{\pi}^{\mu} \cdot \hat{\rho}^{\mu}}{\hat{\pi}^{\mu} + \hat{\rho}^{\mu}}\end{aligned} \tag{5.2}$$

where TP_i denotes the number of *true positives*, FP_i the number of *false positives*, FN_i, the number of *false negatives*, and d is the size of the collection.

5.6.1 Mention Detection

We compared the scores (*precision*/*recall* and *F-measure*) with both tagME and CoreNLP. Wikisim mostly obtains the best precision (outperforming CoreNLP is no surprise as we use it internally). TagME has a better recall, especially on Wiki5000. However, we have a better F1 score on most of the datasets.

5.6.2 Disambiguator

To evaluate the disambiguator as an isolated system, we assume that the mentions are already marked and we only need to link them to the correct entity. We evaluate different modules individually and finally report the results for the *learning to rank* system trained on the combination of them. It is clear from the results that the learned model is successful in combining various statistics and textual-structural information. As we can see, NoPop is the most challenging datasets. Given a dataset with already detected mentions, precision and recall are the same and therefore we only report one number in the Table 5.1.

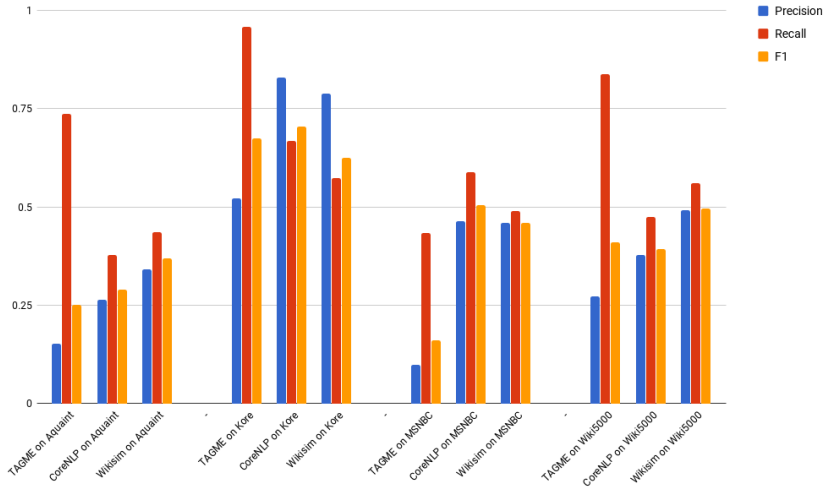


Figure 5.2: Comparison of the *macro scores* of different mention detection methods across different datasets

	Kore	**Aquaint**	**MSNBC**	**Wiki5000**	**NoPop**
Popular	.39	.83	.66	.85	0
Context1	.31	.68	.59	.76	.07
Context2	.5	.75	.65	.86	.1
Word2Vec	.13	.39	.4	.46	.11
Coherence	.5	.51	.62	.73	.21
All (learning to rank Model)	**.57**	**.84**	**.73**	**.91**	**.22**

Table 5.1: Comparing the micro precision of several features and the learned model

5.6.3 Wikifier

The final results for the complete Wikifier are presented in Table 5.2. Our solution achieves comparable results and outperform TagME on the 3 out of the 5 datasets, including the Wiki5000 dataset, which is several times larger than the other datasets. Also the difference on the NoPop dataset is surprising as we obtain a 300% improvement over tagME.

		Kore	Aquaint	MSNBC	Wiki5000	NoPop
Precision	TAGME	**.74**	.3	**.61**	.37	.08
	Wikisim	.63	**.26**	.4	**.47**	**.11**
Recall	TAGME	.23	**.44**	.41	**.64**	**.17**
	Wikisim	**.31**	.42	**.43**	.55	.16
F1	TAGME	.35	**.36**	**.49**	.47	.11
	Wikisim	**.42**	.32	.41	**.51**	**.13**

Table 5.2: Comparing the micro scores of Wikisim with tagME

5.7 Conclusion and Ideas for Further Improvement

We introduced a complete Wikifier system that uses several aspects of text and structure of Wikipedia. We showed that the *key entity* coherence measure, that uses only the structure of Wikipedia, can be combined with a simple context similarity and some other statistics to make a powerful Wikifier. The combination is done using LambdaMart, a list based learning to rank algorithm. The Wikifier uses our proposed entity recognizer, which is built on top of FST dictionary matching and the CoreNLP entity recognizer. The results demonstrate the overall success of our system when compared to the well known entity linker tagME. We also evaluated on a proposed hard-to-disambiguate dataset, i.e., a dataset such that the correct entities are not the most popular ones, and observed that although our system significantly outperforms the baseline, both systems fail to achieve an acceptable F1 score. We believe that the bottleneck of the system is the entity recognizer and conjecture that this task cannot be done independently from the disambiguation. Also, more features and larger scale training (compared to our small training size) may result in a better system.

Chapter 6

Conclusion

We presented a vector space representation of concepts using the graph structure of Wikipedia. The representation is based on the assumption that a concept in Wikipedia is defined by its neighbourhood, i.e, by the its incoming and outgoing neighbours, and all edges between such nodes. We then evaluated several approaches to represent this graph with a vector and demonstrated that *Reversed Pagerank* achieves the best performance in several tasks.

Our main task was evaluating the semantic relatedness quality of the vectors. We compared our proposed method with various graph-based and distributional methods. Among our baselines were several bibliometric and graph overlap metrics, Wikipedia Link Measure (WLM), a simplified Normalized Graph Edit Distance (NGED), different versions of word2vec, Normalized Google Distance (NGD) and word2vec inspired *node embedding*. The results of our thorough evaluations confirmed the quality of our representations.

We also reported the results of incorporating our concept representation in a microblog filtering system [118, 119]. It was used in the query expansion phase of a Twitter Information Filtering (TIF) task, where several other embedding and semantic relatedness methods were compared. The results demonstrated that our vectors performance is superior to the other evaluated methods, which provides more basis for quality of our vectors.

Another outcome of this research was verifying the suitability of Wikipedia in domain-specific semantic relatedness. We compared the performance of our vectors on the biomedical domain datasets with the results obtained from various ontology and distributional methods. Biomedical domain has the highest quality ontologies and corpora, and there exist several successful semantic relatedness methods that served as our baselines. Moreover, we proposed a distributional method applied *word2vec* to a pre-normalized medical corpus using the specialized ontologies. This method achieved the highest performance in the literature on some of the datasets and was a strong baseline for Wikipedia, acknowledged by other studies as well [149, 128]. Wikipedia-based methods were competitive, and in most of the cases, significantly outperformed the on most of the datasets.

We also evaluated our embeddings in the more complex task of word sense disambiguation. We first tried to evaluate our semantic relatedness method in a *coherence* model, that is an optimization that maximizes the sum of pairwise similarities. The advantage of this model is being relatedness neutral. The coherence was modeled using integer linear programming and our semantic relatedness method outperformed other relatedness methods on several datasets. The complexity of the *coherence method* motivated us to find an alternative formulation. We proposed a different notion of coherency by conjecturing that in a short sentence, there is one entity that plays a central role and can disambiguate every other entity. We called this entity *key entity* and redefined coherence to be the sum of the similarities of all entities to this *key entity*. This new optimization can be solved by simply searching in quadratic time and when solved, can improve the results.

In our next attempt, we showed that having vector representation can help to find the *key entity* in linear time. This was done by introducing a *Vector Space Model* (VSM) based way to model context, that is benefiting from vector space operations, such as adding and subtracting vectors. The new algorithm showed to be more effective, both in terms of accuracy and computation time.

We finally tried to combine our graph based method with the text of Wikipedia, which is another important feature of Wikipedia that we had previously ignored. We combined our structure-based *coherence* with a text-based *context similarity* (and also other useful statistics, such as popularity and the prior probabilities) in our WSD approach. We used a *learning-to-rank* algorithm and the learned model denoted the success of the combination. We built a complete Wikifier on this model and provided an open-source software to be used in further research [1].

6.1 Possible Extensions

Following the structure of the thesis, there are several potential avenues with which one could extend our research.

[1] `https://github.com/asajadi/wikisim`

6.1.1 Low-Dimensional Embedding of Our Representations

Starting from the embeddings, one natural extension to our research is to investigate a lower dimensional representation of our embeddings. We showed that a traditional matrix factorization of the adjacency graph is underperforming compared to the local embedding method. Therefore, another round of embedding on the local embeddings should give us the advantages of dense embedding and our sparse representation at the same time. Global embedding can also fail because the adjacency might not be enough to reflect similarity. In this case, trying to use higher orders of proximity, similar to the approaches taken by [195, 198] may result in a successful dense embedding of Wikipedia concepts.

6.1.2 Knowledge Graph Embedding

Our proposed concept representation can be evaluated on knowledge graphs and ontologies, such as Wikidata [2]. Wikidata is a free, collaboratively edited and structured database. It contains an extensive collection of data, in an (*item*, *property*, *value*) format, for example: (*Douglas Adams, educated at*, *St John's College*). Wikidata is the collaboratively edited equivalent of Google's *Knowledge Graph* [3] or Max-Planck's *YAGO* [4]. The triplets form a labelled graph with two types of nodes: *items* and *attributes*, connected with edges labelled with the *properties*.

Wikidata has several advantages over Wikipedia which may result in more quality embeddings, such as:

- The structure is less noisy. A concept is only linked to objects that have a well-defined relation with.

- All relations are labelled, so even more pruning is possible. A manual or learned method can select relations that are contributing more to the definition of the object, prior to the embedding.

- Since the neighbours are all labelled, it is possible to have several embeddings for a concept, each representing one aspect of it. For example, a diverse concept like

[2] https://www.wikidata.org

[3] https://www.google.com/intl/bn/insidesearch/features/search/knowledge.html

[4] http://www.yago-knowledge.org/

Canada can have multiple embeddings reflecting its *political*, *historical* or *geographical* information.

6.1.3 Multiple Key Entity

Regarding WSD and *key entity*, we assumed the existence of only one *key entity*, while it is only logical to assume there would be more than one. Thinking of these entities as the *focal point*s of a text, it means that there are multiple entities in a text which collectively define the context, or even, each of them individually can help disambiguating a different set of entities. Multiple key entities change the coherence definition again and require different strategies to be found.

6.1.4 Word Embedding for WSD

Another surprise with *word2vec* was its poor performance in WSD. There exist several approaches in the literature to embed different senses of a word, known as multi-sense, or multi-prototype embedding [35, 204, 17, 129, 121]. These approaches are very similar to the one analyzed in this study, $word2vec_2$ (sec. 3.8.1). Our evaluations showed that embedding different sense is not necessarily useful in word sense disambiguation. However, word embedding can be trained specifically for word sense disambiguation, that is, assigning vectors to senses in a way that they are close to their context and far from the context of the other senses. This second constraint is the main difference between similar studies that we have mentioned and can be satisfied through *negative sampling*.

6.1.5 Joint Mention Detection/Disambiguation

And finally, a possible extension to our wikification is to improve the quality of the mention detection. Traditionally, mention detection is done prior to disambiguation as a separate phase. Our experiments showed that using a more complex approach that jointly detects and disambiguates mentions is the only way to improve the quality of the mention detector. The simplest solution can be using our *Finite State Transducer* approach to detect all possible overlapping mentions, combining all the candidates for overlapping mentions and performing the disambiguation. At the end, among each overlapping group, the one that its entity was selected as the target would be the correct mention.

6.1.6 Multiple Knowledge-Source Embedding and Linking

While Wikipedia has a good coverage on special domains, it has a very limited information about general words, terms used in social media (daily evolving hashtags and abbreviations), news (named entities not necessarily in Wikipedia) and personal (private) domains, such as a terminology used in a private corporation. We also mentioned that general thesauri, such as Wordnet and knowledge graphs, such as Freebase or Wikidata, can contain complementary knowledge to Wikipedia. One important direction to persuade is to use several entity linkers (our Wikisim, or Tulip [114] which can link to Freebase) and link a text to multiple resources. We can then combine the information from all of these resources to analyse the text. This can be done either by simple approaches, such as the way we used *learning to rank* to combine the structure and the text of Wikipedia in Chapter 5, or even more fundamental approaches, such embedding all of the knowledge bases in the same space (co-embedding).

Bibliography

[1] Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor 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*, NAACL '09, pages 19–27, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[2] Eneko Agirre, Ander Barrena, and Aitor Soroa. Studying the wikipedia hyperlink graph for relatedness and disambiguation. *CoRR*, abs/1503.01655, 2015.

[3] Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-agirre, and Weiwei Guo. SEM 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity. In **SEM 2013: The Second Joint Conference on Lexical and Computational Semantics. Association for Computational Linguistics*, 2013.

[4] Eneko Agirre, Montse Cuadros, German Rigau, and Aitor Soroa. Exploring knowledge bases for similarity. In *LREC*, 2010.

[5] Eneko Agirre and Philip Edmonds. *Word Sense Disambiguation: Algorithms and Applications*. Springer Publishing Company, Incorporated, 1st edition, 2007.

[6] Eneko Agirre, Oier López de Lacalle, and Aitor Soroa. Random walks for knowledge-based word sense disambiguation. *Comput. Linguist.*, 40(1):57–84, March 2014.

[7] Eneko Agirre, David Martínez, Oier López de Lacalle, and Aitor Soroa. Two graph-based algorithms for state-of-the-art wsd. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 585–593, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[8] Eneko Agirre and Aitor Soroa. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 33–41, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[9] Eneko Agirre, Aitor Soroa, and Mark Stevenson. Graph-based word sense disambiguation of biomedical documents. *Bioinformatics*, 26(22):2889–2896, November 2010.

[10] Alfred V. Aho and Margaret J. Corasick. Efficient string matching: An aid to bibliographic search. *Commun. ACM*, 18(6):333–340, June 1975.

[11] R.A. Amsler. *Applications of Citation-based Automatic Classification*. Internal technical report - Linguistics Research Center, University of Texas at Austin. Linguistics Research Center, University of Texas at Austin, 1972.

[12] Alan R. Aronson and Franois-Michel Lang. An overview of metamap: historical perspective and recent advances. *JAMIA*, 17(3):229–236, 2010.

[13] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval - the concepts and technology behind search, Second edition*. Pearson Education Ltd., Harlow, England, 2011.

[14] Satanjeev Banerjee and Ted Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing: Third International Conference, CICLing 2002 Mexico City, Mexico, February 17–23, 2002 Proceedings*, pages 136–145. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[15] Satanjeev Banerjee and Ted Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *IJCAI*, pages 805–810, 2003.

[16] Ziv Bar-Yossef and Li-Tal Mashiach. Local approximation of pagerank and reverse pagerank. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 279–288, New York, NY, USA, 2008. ACM.

[17] Sergey Bartunov, Dmitry Kondrashkin, Anton Osokin, and Dmitry P. Vetrov. Breaking sticks and ambiguities with adaptive skip-gram. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, pages 130–138, 2016.

[18] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pages 585–591, Cambridge, MA, USA, 2001. MIT Press.

[19] Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. Tabel: Entity linking in web tables. In *The Semantic Web - ISWC 2015: 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I*, pages 425–441. Springer International Publishing, Cham, 2015.

[20] Vincent D. Blondel, Anahí Gajardo, Maureen Heymans, Pierre Senellart, and Paul Van Dooren. A measure of similarity between graph vertices: Applications to synonym extraction and web searching. *SIAM Rev.*, 46(4):647–666, April 2004.

[21] Olivier Bodenreider. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(Database-Issue):267–270, 2004.

[22] D. Bollegala, Y. Matsuo, and M. Ishizuka. A web search engine-based approach to measure semantic similarity between words. *Knowledge and Data Engineering, IEEE Transactions on*, 23(7):977–990, 2011.

[23] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2787–2795. Curran Associates, Inc., 2013.

[24] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, pages 301–306. AAAI Press, 2011.

[25] Alexander Budanitsky. *Lexical Semantic Relatedness and its Application in Natural Language Processing*. PhD thesis, University of Toronto, Toronto, Ontario, 1999.

[26] Razvan Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proceesings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, pages 9–16, Trento, Italy, 2006.

[27] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 89–96, New York, NY, USA, 2005. ACM.

[28] Chris J.C. Burges. From ranknet to lambdarank to lambdamart: An overview. Technical report, June 2010.

[29] Chris J.C. Burges, Robert Ragno, and Quoc Viet Le. Learning to rank with non-smooth cost functions. In *Advances in Neural Information Processing Systems 19*, January 2007.

[30] Jorge E. Caviedes and James J. Cimino. Towards the development of a conceptual distance metric for the umls. *J. of Biomedical Informatics*, 37(2):77–85, April 2004.

[31] Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. Learning relatedness measures for entity linking. In *Proceedings of the 22Nd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '13, pages 139–148, New York, NY, USA, 2013. ACM.

[32] Ming-Wei Chang, Lev Ratinov, and Dan Roth. Structured learning with constrained conditional models. *Mach. Learn.*, 88(3):399–431, September 2012.

[33] Ming-Wei Chang, Nicholas Rizzolo, and Dan Roth. Integer linear programming in nlp - constrained conditional models. In *NAACL HLT 2010 Tutorial Abstracts*, pages 9–14, Los Angeles, California, June 2010. Association for Computational Linguistics.

[34] Chandra Chekuri, Sanjeev Khanna, Joseph (Seffi) Naor, and Leonid Zosin. Approximation algorithms for the metric labeling problem via a new linear programming formulation. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '01, pages 109–118, Philadelphia, PA, USA, 2001. Society for Industrial and Applied Mathematics.

[35] Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. A unified model for word sense representation and disambiguation. 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 1025–1035, 2014.

[36] Xiao Cheng and Dan Roth. Relational inference for wikification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1787–1796, 2013.

[37] Andrew Chisholm and Ben Hachey. Entity disambiguation with web links. *Transactions of the Association for Computational Linguistics*, 3:145–156, 2015.

[38] David Christensen. Fast algorithms for the calculation of Kendall's τ. *Computational Statistics*, 20(1):51–62, 2005.

[39] Rudi L. Cilibrasi and Paul M. B. Vitanyi. The google similarity distance. *IEEE Trans. on Knowl. and Data Eng.*, 19(3):370–383, March 2007.

[40] K. A. Clauson, H. H. Polen, M. N. Boulos, and J. H. Dzenowagis. Scope, completeness, and accuracy of drug information in Wikipedia. *Ann Pharmacother*, 42(12):1814–1821, Dec 2008.

[41] Donatello Conte, Jean-Yves Ramel, Nicolas Sidère, Muhammad Muzzamil Luqman, Benoît Gaüzère, Jaume Gibert, Luc Brun, and Mario Vento. A comparison of explicit and implicit graph embedding methods for pattern recognition. In *Graph-Based Representations in Pattern Recognition: 9th IAPR-TC-15 International Workshop, GbRPR 2013, Vienna, Austria, May 15-17, 2013. Proceedings*, pages 81–90. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[42] Marco Cornolti, Paolo Ferragina, Massimiliano Ciaramita, Stefan Rüd, and Hinrich Schütze. A piggyback system for joint entity mention detection and linking in web queries. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 567–578, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee.

[43] Thierson Couto, Marco Cristo, Marcos André Gonçalves, Pável Calado, Nivio Ziviani, Edleno Moura, and Berthier Ribeiro-Neto. A comparative study of citations and links in document classification. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '06, pages 75–84, New York, NY, USA, 2006. ACM.

[44] Silviu Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[45] Ronald Fagin, Ravi Kumar, and D. Sivakumar. Comparing top k lists. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '03, pages 28–36, Philadelphia, PA, USA, 2003. Society for Industrial and Applied Mathematics.

[46] Paolo Ferragina and Ugo Scaiella. Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1625–1628, New York, NY, USA, 2010. ACM.

[47] Miroslav Fiedler. Laplacian of graphs and algebraic connectivity. *Banach Center Publications*, 25(1):57–70, 1989.

[48] E. C. Fieller, H. O. Hartley, and E. S. Pearson. Tests for rank correlation coefficients. i. *Biometrika*, 44(3/4):pp. 470–481, 1957.

[49] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[50] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: the concept revisited. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 406–414, New York, NY, USA, 2001. ACM.

[51] Dániel Fogaras. Where to start browsing the web? In Thomas Böhme, Gerhard Heyer, and Herwig Unger, editors, *Innovative Internet Community Systems: Third International Workshop, IICS 2003, Leipzig, Germany, June 19-21, 2003. Revised Papers*, pages 65–79. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[52] John Forrest and Robin Lougee-Heimer. Cbc user guide. In *Emerging Theory, Methods, and Applications*, chapter Chapter 10, pages 257–277. INFORMS, 2005.

[53] Matthew Francis-Landau, Greg Durrett, and Dan Klein. Capturing semantic similarity for entity linking with convolutional neural networks. In *Proceedings of the North American Association for Computational Linguistics*, San Diego, California, USA, June 2016. Association for Computational Linguistics.

[54] Ana Freire, Matteo Manca, Diego Saez-Trumper, David Laniado, Ilaria Bordino, Francesco Gullo, and Andreas Kaltenbrunner. Graph-based breaking news detection on wikipedia. *Biography*, 6:1, 2016.

[55] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

[56] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, IJCAI'07, pages 1606–1611, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[57] Octavian-Eugen Ganea, Marina Ganea, Aurelien Lucchi, Carsten Eickhoff, and Thomas Hofmann. Probabilistic bag-of-hyperlinks model for entity linking. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 927–938, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee.

[58] Xinbo Gao, Bing Xiao, Dacheng Tao, and Xuelong Li. A survey of graph edit distance. *Pattern Analysis and Applications*, 13(1):113–129, 2010.

[59] Peter Gärdenfors. *Conceptual spaces: The geometry of thought*. MIT press, 2004.

[60] Vijay N. Garla and Cynthia Brandt. Knowledge-based biomedical word sense disambiguation: An evaluation and application to clinical document classification. In *Proceedings of the 2012 IEEE Second International Conference on Healthcare Informatics, Imaging and Systems Biology*, HISB '12, pages 22–, Washington, DC, USA, 2012. IEEE Computer Society.

[61] VijayN Garla and Cynthia Brandt. Semantic similarity in the biomedical domain: an evaluation across knowledge sources. *BMC Bioinformatics*, 13(1):1–13, 2012.

[62] P. Garrard, M. A. Ralph, J. R. Hodges, and K. Patterson. Prototypicality, distinctiveness, and intercorrelation: Analyses of the semantic attributes of living and nonliving concepts. *Cognitive Neuropsychology*, 18(2):125–174, March 2001.

[63] Alexander F. Gelbukh, editor. *Computational Linguistics and Intelligent Text Processing - 16th International Conference, CICLing 2015, Cairo, Egypt, April 14-20, 2015, Proceedings, Part I*, volume 9041 of *Lecture Notes in Computer Science*. Springer, 2015.

[64] Jaume Gibert, Ernest Valveny, and Horst Bunke. Feature selection on node statistics based embedding of graphs. *Pattern Recognition Letters*, 33(15):1980 – 1990, 2012. Graph-Based Representations in Pattern Recognition.

[65] Jaume Gibert, Ernest Valveny, and Horst Bunke. Graph embedding in vector spaces by node attribute statistics. *Pattern Recognition*, 45(9):3072 – 3083, 2012. Best Papers of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA'2011).

[66] Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *CoRR*, abs/1402.3722, 2014.

[67] Dan Goldwasser, Vivek Srikumar, and Dan Roth. Predicting structures in nlp: Constrained conditional models and integer linear programming nlp. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials*, NAACL HLT '12, pages 8:1–8:4, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[68] Gene H. Golub and Henk A. van der Vorst. Eigenvalue computation in the 20th century. *Journal of Computational and Applied Mathematics*, 123(1-2):35 – 65, 2000. Numerical Analysis 2000. Vol. III: Linear Algebra.

[69] Thomas L. Griffiths, Mark Steyvers, and Joshua B. Tenenbaum. Topics in semantic representation. *Psychological Review*, 114(2):211–244, 2007.

[70] Joan Guisado-Gámez, David Tamayo-Domenech, Jordi Urmeneta, and Josep Lluís Larriba-Pey. Enrich: A query rewriting service powered by wikipedia graph structure. In *Tenth International AAAI Conference on Web and Social Media*, 2016.

[71] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, pages 576–587. VLDB Endowment, 2004.

[72] Lushan Han, Abhay L. Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. Umbc_ebiquity-core: Semantic textual similarity systems. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics, *SEM 2013, June 13-14, 2013, Atlanta, Georgia, USA.*, pages 44–52, 2013.

[73] Xianpei Han and Jun Zhao. Named entity disambiguation by leveraging wikipedia semantic knowledge. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 215–224, New York, NY, USA, 2009. ACM.

[74] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer New York, 2009.

[75] Larry Heck and Hongzhao Huang. Deep learning of knowledge graph embeddings for semantic parsing of twitter dialogs. In *2014 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2014, Atlanta, GA, USA, December 3-5, 2014*, pages 597–601, 2014.

[76] William Hersh, Chris Buckley, T. J. Leone, and David Hickam. Ohsumed: An interactive retrieval evaluation and new large test collection for research. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pages 192–201, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[77] Birger Hjrland. Citation analysis: A social and dynamic approach to knowledge organization. *Information Processing & Management*, 49(6):1313–1325, November 2013.

[78] Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. Kore: Keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 545–554, New York, NY, USA, 2012. ACM.

[79] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artif. Intell.*, 194:28–61, January 2013.

[80] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 782–792, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[81] Hongzhao Huang, Larry Heck, and Heng Ji. Leveraging deep neural networks and knowledge graphs for entity disambiguation. *CoRR*, abs/1504.07678, 2015.

[82] G. Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, 14(1):55–63, January 1968.

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

[84] Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.

[85] Nancy Ide and Jean Véronis. Introduction to the special issue on word sense disambiguation: The state of the art. *Comput. Linguist.*, 24(1):2–40, March 1998.

[86] Elias Iosif and Alexandros Potamianos. Unsupervised semantic similarity computation between terms using web documents. *IEEE Trans. on Knowl. and Data Eng.*, 22(11):1637–1647, November 2010.

[87] Aminul Islam, Evangelos E. Milios, and Vlado Keselj. Text similarity using google tri-grams. In Leila Kosseim and Diana Inkpen, editors, *Canadian Conference on AI*, volume 7310 of *Lecture Notes in Computer Science*, pages 312–317. Springer, 2012.

[88] Shahida Jabeen, Xiaoying Gao, and Peter Andreae. CPRel: Semantic relatedness computation using wikipedia based context profiles. In *Research in Computing Science*, volume 70, pages 55–66, 2013.

[89] Glen Jeh and Jennifer Widom. Simrank: a measure of structural-context similarity. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 538–543, New York, NY, USA, 2002. ACM.

[90] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *CoRR*, cmp-lg/9709008, 1997.

[91] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2000.

[92] Mikael Kågebäck, Fredrik D. Johansson, Richard Johansson, and Devdatt P. Dubhashi. Neural context embeddings for automatic discovery of word senses. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, VS@NAACL-HLT 2015, June 5, 2015, Denver, Colorado, USA*, pages 25–32, 2015.

[93] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[94] M. M. Kessler. Bibliographic coupling between scientific papers. *Amer. Doc.*, 14(1):10–25, 1963.

[95] A. Kilgarriff and J. Rosenzweig. Framework and results for english senseval. *Computers and the Humanities*, 34(1):15–48, 2000.

[96] Jon Kleinberg and Eva Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, FOCS '99, pages 14–, Washington, DC, USA, 1999. IEEE Computer Society.

[97] Jon Kleinberg and Éva Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. *J. ACM*, 49(5):616–639, September 2002.

[98] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, September 1999.

[99] Bevan Koopman, Guido Zuccon, Peter Bruza, Laurianne Sitbon, and Michael Lawley. An evaluation of corpus-driven measures of medical concept similarity for information retrieval. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 2439–2442, New York, NY, USA, 2012. ACM.

[100] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 457–466, New York, NY, USA, 2009. ACM.

[101] Thomas K Landauer and Susan T. Dutnais. A solution to platos problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, pages 211–240, 1997.

[102] Thomas K Landauer, Peter W. Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25(2-3):259–284, 1998.

[103] Nevena Lazic, Amarnag Subramanya, Michael Ringgaard, and Fernando Pereira. Plato: A selective context model for entity resolution. *TACL*, 3:503–515, 2015.

[104] C. Leacock and M. Chodorow. Combining local context and wordnet similarity for word sense identification. In *WordNet: An Electronic Lexical Database*, pages 305–332. In C. Fellbaum (Ed.), MIT Press, 1998.

[105] Joonseok Lee, Ariel Fuxman, Bo Zhao, and Yuanhua Lv. Leveraging knowledge bases for contextual entity exploration. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1949–1958, New York, NY, USA, 2015. ACM.

[106] Lee:2002. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 41–48, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[107] Michael 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*, SIGDOC '86, pages 24–26, New York, NY, USA, 1986. ACM.

[108] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Curran Associates, Inc., 2014.

[109] Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015.

[110] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, CIKM '03, pages 556–559, New York, NY, USA, 2003. ACM.

[111] Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[112] Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou. Identifying synonyms among distributionally similar words. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, pages 1492–1493, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc.

[113] Jimmy Lin, Miles Efron, Yulu Wang, Garrick Sherman, , and Ellen Voorhees. Overview of the TREC-2015 microblog track. In *Proceedings of The Twenty-Fourth Text REtrieval Conference, TREC 2015, Gaithersburg, Maryland, USA, November 17-20, 2015*, 2015.

[114] Marek Lipczak, Arash Koushkestani, and Evangelos E. Milios. Tulip: lightweight entity recognition and disambiguation using wikipedia-based topic centroids. In *ERD'14, Proceedings of the First ACM International Workshop on Entity Recognition & Disambiguation, July 11, 2014, Gold Coast, Queensland, Australia*, pages 31–36, 2014.

[115] Tie-Yan Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331, March 2009.

[116] R. Lougee-Heimer. The common optimization interface for operations research: Promoting open-source software in the operations research community. *IBM J. Res. Dev.*, 47(1):57–66, January 2003.

[117] Wangzhong Lu, J. Janssen, E. Milios, N. Japkowicz, and Yongzheng Zhang. Node similarity in the citation graph. *Knowledge and Information Systems*, 11(1):105–129, 2007.

[118] R. Makki, A. J. Soto, S. Brooks, and E. E. Milios. Twitter message recommendation based on user interest profiles. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 406–410, Aug 2016.

[119] Raheleh Makki. *Interactive Text Analytics For User-Generated Content*. PhD thesis, Dalhousie University, Halifax, Canada, 2017.

[120] John C. Mallery. Thinking about foreign policy: Finding an appropriate role for artificially intelligent computers. In *Master's thesis, M.I.T. Political Science Department*, 1988.

[121] Massimiliano Mancini, José Camacho-Collados, Ignacio Iacobacci, and Roberto Navigli. Embedding words and senses together via joint knowledge-enhanced training. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*, pages 100–111, 2017.

[122] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[123] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[124] Eric Margolis and Stephen Laurence. Concepts. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. The Metaphysics Research Lab Center for the Study of Language and Information Stanford University Stanford, CA 94305-4115, spring 2014 edition, 2014.

[125] Lluís Màrquez, Gerard Escudero, David Martínez, and German Rigau. Supervised corpus-based methods for wsd. In Eneko Agirre and Philip Edmonds, editors, *Word Sense Disambiguation: Algorithms and Applications*, pages 167–216. Springer Netherlands, Dordrecht, 2006.

[126] B. T. McInnes, T. Pedersen, and S. V. Pakhomov. UMLS-Interface and UMLS-Similarity : open source software for measuring paths and semantic similarity. *AMIA Annual Symposium Proc*, 2009:431–435, 2009.

[127] Bridget T. McInnes and Ted Pedersen. Evaluating measures of semantic similarity and relatedness to disambiguate terms in biomedical text. *J. of Biomedical Informatics*, 46(6):1116–1124, December 2013.

[128] Bridget T. McInnes and Ted Pedersen. Improving correlation with human judgments by integrating semantic similarity with second-order vectors. In *BioNLP 2017, Vancouver, Canada, August 4, 2017*, pages 107–116, 2017.

[129] Oren Melamud, Jacob Goldberger, and Ido Dagan. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61. Association for Computational Linguistics, 2016.

[130] Rada Mihalcea. Knowledge-based methods for wsd. In Eneko Agirre and Philip Edmonds, editors, *Word Sense Disambiguation: Algorithms and Applications*, pages 107–131. Springer Netherlands, Dordrecht, 2006.

[131] Rada Mihalcea and Andras Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 233–242, New York, NY, USA, 2007. ACM.

[132] Rada F. Mihalcea and Dragomir R. Radev. *Graph-based Natural Language Processing and Information Retrieval*. Cambridge University Press, New York, NY, USA, 1st edition, 2011.

[133] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[134] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3111–3119, 2013.

[135] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[136] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995.

[137] George A. Miller and Walter G. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28, 1991.

[138] David Milne and Ian H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 509–518, New York, NY, USA, 2008. ACM.

[139] David Milne and Ian H. Witten. An open-source toolkit for mining wikipedia. *Artif. Intell.*, 194:222–239, January 2013.

[140] Cleve Moler. *Experiments with MATLAB = MATLAB zhi fu : bian cheng shi jian*. Beijing hang kong hang tian da xue chu ban she, Beijing Shi, 2013.

[141] Brian Murphy, Partha Pratim Talukdar, and Tom M. Mitchell. Learning effective and interpretable semantic models using non-negative sparse embedding. In *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India*, pages 1933–1950, 2012.

[142] Katta Murty. *Linear programming*. John Wiley & Sons, New York, 1983.

[143] Roberto Navigli. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2):10:1–10:69, February 2009.

[144] Roberto Navigli and Mirella Lapata. An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(4):678–692, April 2010.

[145] H.A. Nguyen and H. Al-Mubaid. New ontology-based semantic similarity measure for the biomedical domain. In *Granular Computing, 2006 IEEE International Conference on*, pages 623–628, 2006.

[146] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999.

[147] S. Pakhomov, B. McInnes, T. Adam, Y. Liu, T. Pedersen, and G. B. Melton. Semantic Similarity and Relatedness between Clinical Terms: An Experimental Study. *AMIA Annu Symp Proc*, 2010:572–576, 2010.

[148] Serguei V. S. Pakhomov, Ted Pedersen, Bridget McInnes, Genevieve B. Melton, Alexander Ruggieri, and Christopher G. Chute. Towards a framework for developing semantic relatedness reference standards. *J. of Biomedical Informatics*, 44(2):251–265, April 2011.

[149] Serguei V.S. Pakhomov, Greg Finley, Reed McEwan, Yan Wang, and Genevieve B. Melton. Corpus domain effects on distributional semantic modeling of medical terms. *Bioinformatics*, 32(23):3635–3644, 2016.

[150] Martha Palmer, Hwee Tou Ng, and Hoa Trang Dang. Evaluation of wsd systems. In Eneko Agirre and Philip Edmonds, editors, *Word Sense Disambiguation: Algorithms and Applications*, pages 75–106. Springer Netherlands, Dordrecht, 2006.

[151] Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. *English Gigaword Fifth Edition LDC2011T07*. Linguistic Data Consortium, Philadelphia, PA, 2011.

[152] Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing'03, pages 241–257, Berlin, Heidelberg, 2003. Springer-Verlag.

[153] Siddharth Patwardhan and Ted Pedersen. Using wordnet-based context vectors to estimate the semantic relatedness of concepts. In *In: Proceedings of the EACL*, pages 1–8, 2006.

[154] Ted Pedersen. Unsupervised corpus-based methods for wsd. In Eneko Agirre and Philip Edmonds, editors, *Word Sense Disambiguation: Algorithms and Applications*, pages 133–166. Springer Netherlands, Dordrecht, 2006.

[155] Ted Pedersen and Rebecca F. Bruce. Distinguishing word senses in untagged text. *CoRR*, cmp-lg/9706008, 1997.

[156] Ted Pedersen, Serguei V.S. Pakhomov, Siddharth Patwardhan, and Christopher G. Chute. Measures of semantic similarity and relatedness in the biomedical domain. *Journal of Biomedical Informatics*, 40(3):288 – 299, 2007.

[157] Maria Pelevina, Nikolay Arefyev, Chris Biemann, and Alexander Panchenko. Making sense of word embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 174–183, 2016.

[158] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[159] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 701–710, New York, NY, USA, 2014. ACM.

[160] Francesco Piccinno and Paolo Ferragina. From tagme to wat: A new entity annotator. In *Proceedings of the First International Workshop on Entity Recognition & Disambiguation*, ERD '14, pages 55–62, New York, NY, USA, 2014. ACM.

[161] Simone Paolo Ponzetto and Michael Strube. Knowledge derived from wikipedia for computing semantic relatedness. *J. Artif. Intell. Res. (JAIR)*, 30:181–212, 2007.

[162] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007.

[163] Paul Procter. *Longman dictionary of contemporary English*. Longman, Harlow England, 1978.

[164] Vasin Punyakanok, Dan Roth, Wen-tau Yih, and Dav Zimak. Semantic role labeling via integer linear programming inference. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[165] Amruta Purandare and Ted Pedersen. Word sense discrimination by clustering contexts in vector and similarity spaces. In *Proceedings of the Eighth Conference on Computational Natural Language Learning, CoNLL 2004, Held in cooperation with HLT-NAACL 2004, Boston, Massachusetts, USA, May 6-7, 2004*, pages 41–48, 2004.

[166] 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.

[167] L. Ratinov and D. Roth. Design challenges and misconceptions in named entity recognition. In *CoNLL*, 6 2009.

[168] Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1375–1384, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[169] N. J. Reavley, A. J. Mackinnon, A. J. Morgan, M. Alvarez-Jimenez, S. E. Hetrick, E. Killackey, B. Nelson, R. Purcell, M. B. Yap, and A. F. Jorm. Quality of information sources about mental disorders: a comparison of Wikipedia with centrally controlled web and printed sources. *Psychol Med*, 42(8):1753–1762, Aug 2012.

[170] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI*, pages 448–453, 1995.

[171] Kaspar Riesen, Michel Neuhaus, and Horst Bunke. Graph embedding in vector spaces by means of prototype selection. In *Graph-Based Representations in Pattern Recognition: 6th IAPR-TC-15 International Workshop, GbRPR 2007, Alicante, Spain, June 11-13, 2007. Proceedings*, pages 383–393. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[172] Xin Rong. word2vec parameter learning explained. *CoRR*, abs/1411.2738, 2014.

[173] Eleanor Rosch. Principles of categorization. In Allan Collins and Edward E. Smith, editors, *Readings in Cognitive Science, a Perspective From Psychology and Artificial Intelligence*, pages 312–22. Morgan Kaufmann Publishers, 1988.

[174] Eleanor Rosch and Carolyn B Mervis. Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, 7:573–605, October 1975.

[175] Dan Roth, Heng Ji, Ming-Wei Chang, and Taylor Cassidy. Wikification and beyond: The challenges of entity and concept grounding. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Tutorial Abstracts*, page 7, 2014.

[176] Dan Roth and Scott Wen-tau Yih. Global inference for entity and relation identification via a linear programming formulation. In *Introduction to Statistical Relational Learning*. MIT Press, November 2007.

[177] Dan Roth and Wen-tau Yih. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning, CoNLL 2004, Held in cooperation with HLT-NAACL 2004, Boston, Massachusetts, USA, May 6-7, 2004*, pages 1–8, 2004.

[178] Herbert Rubenstein and John B. Goodenough. Contextual correlates of synonymy. *Commun. ACM*, 8(10):627–633, October 1965.

[179] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986.

[180] Armin Sajadi, Evangelos E. Milios, and Vlado Keselj. Vector space representation of concepts using wikipedia graph structure. In Flavius Frasincar, Ashwin Ittoo, Le Minh Nguyen, and Elisabeth Métais, editors, *Natural Language Processing and Information Systems: 22nd International Conference on Applications of Natural Language to Information Systems, NLDB 2017, Liège, Belgium, June 21-23, 2017, Proceedings*, pages 393–405. Springer International Publishing, Cham, 2017.

[181] Armin Sajadi, Evangelos E. Milios, Vlado Kešelj, and Jeannette C. M. Janssen. Domain-specific semantic relatedness from wikipedia structure: A case study in biomedical text. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing: 16th International Conference, CICLing 2015, Cairo, Egypt, April 14-20, 2015, Proceedings, Part I*, pages 347–360. Springer International Publishing, Cham, 2015.

[182] David Sánchez and Montserrat Batet. Semantic similarity estimation in the biomedical domain: An ontology-based information-theoretic perspective. *J. of Biomedical Informatics*, 44(5):749–759, October 2011.

[183] Ugo Scaiella, Paolo Ferragina, Andrea Marino, and Massimiliano Ciaramita. Topical clustering of search results. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, pages 223–232, New York, NY, USA, 2012. ACM.

[184] Hinrich Schütze. Automatic word sense discrimination. *Comput. Linguist.*, 24(1):97–123, March 1998.

[185] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, March 2002.

[186] Nuno Seco, Tony Veale, and Jer Hayes. An intrinsic information content metric for semantic similarity in wordnet. In *ECAI*, pages 1089–1090, 2004.

[187] Pierre Senellart and Vincent D. Blondel. Automatic discovery of similar words. In Michael W. Berry and Malu Castellanos, editors, *Survey of Text Mining II: Clustering, Classification and Retrieval*, pages 25–44. Springer-Verlag, January 2008.

[188] Ehsan Sherkat and Evangelos E. Milios. Vector embedding of wikipedia concepts and entities. In *Natural Language Processing and Information Systems: 22nd International Conference on Applications of Natural Language to Information Systems, NLDB 2017, Liège, Belgium, June 21-23, 2017, Proceedings*, pages 418–428. Springer International Publishing, Cham, 2017.

[189] N. Sidre, P. Hroux, and J. Y. Ramel. Vector representation of graphs: Application to the classification of symbols and letters. In *2009 10th International Conference on Document Analysis and Recognition*, pages 681–685, July 2009.

[190] H. Small. Co-citation in scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for Information Science (JASIS)*, 24(4):265–269, 1973.

[191] Marina Sokolova and Peter van Beek, editors. *Advances in Artificial Intelligence - 27th Canadian Conference on Artificial Intelligence, Canadian AI 2014, Montréal, QC, Canada, May 6-9, 2014. Proceedings*, volume 8436 of *Lecture Notes in Computer Science*. Springer, 2014.

[192] Daniel A Spielman. Spectral graph theory lecture 2: The laplacian. `http://www.cs.yale.edu/homes/spielman/561/lect02-15.pdf`, 2015. "[Online; accessed 2017-October-20]".

[193] Daniel A. Spielman and Shang-Hua Teng. Spectral partitioning works: Planar graphs and finite element meshes. *Linear Algebra and its Applications*, 421(23):284305, Mar 2007.

[194] Michael Symonds, Guido Zuccon, Bevan Koopman, Peter D. Bruza, and Anthony Nguyen. Semantic judgement of medical concepts : combining syntagmatic and paradigmatic information with the tensor encoding model. In *Australasian Language Technology Association Workshop (ALTA 2012)*, University of Otago, Dunedin, December 2012.

[195] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 1067–1077, Republic and Canton of Geneva, Switzerland, 2015. International World Wide Web Conferences Steering Committee.

[196] Florentina Vasilescu, Philippe Langlais, and Guy Lapalme. Evaluating variants of the lesk approach for disambiguating words. In *Proceedings of LREC 2004*, pages 633–636, Lisbonne, may 2004.

[197] Jean Vronis. Hyperlex: lexical cartography for information retrieval. *Computer Speech & Language*, 18(3):223 – 252, 2004. Word Sense Disambiguation.

[198] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1225–1234, New York, NY, USA, 2016. ACM.

[199] Dirk Weissenborn, Feiyu Xu, and Hans Uszkoreit. DFKI: multi-objective optimization for the joint disambiguation of entities and nouns & deep verb sense disambiguation. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, pages 335–339, 2015.

[200] Robert West, Ashwin Paranjape, and Jure Leskovec. Mining missing hyperlinks from human navigation traces: A case study of wikipedia. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 1242–1252, New York, NY, USA, 2015. ACM.

[201] Richard C. Wilson, Edwin R. Hancock, and Bin Luo. Pattern vectors from algebraic graph theory. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(7):1112–1124, July 2005.

[202] Ian Witten and David Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceedings of AAAI 2008*, pages 25–30, 2008.

[203] A. Wood and K. Struthers. Pathology education, Wikipedia and the Net generation. *Med Teach*, 32(7):618, 2010.

[204] Zhaohui Wu and C. Lee Giles. Sense-aware semantic analysis: A multi-prototype word representation model using wikipedia. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2188–2194. AAAI Press, 2015.

[205] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

[206] Beibei Yang and Jesse M. Heines. Domain-specific semantic relatedness from Wikipedia: can a course be transferred? In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*, NAACL HLT '12, pages 35–40, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[207] David Yarowsky. One sense per collocation. In *Proceedings of the Workshop on Human Language Technology*, HLT '93, pages 266–271, Stroudsburg, PA, USA, 1993. Association for Computational Linguistics.

[208] Majid Yazdani and Andrei Popescu-Belis. Computing text semantic relatedness using the contents and links of a hypertext encyclopedia. *Artif. Intell.*, 194:176–202, 2013.

[209] Eric Yeh, Daniel Ramage, Christopher D. Manning, Eneko A, and Aitor Soroa. Wikiwalk: random walks on Wikipedia for semantic relatedness. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, TextGraphs-4, pages 41–49, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[210] Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. Semi-supervised word sense disambiguation with neural models. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 1374–1385, 2016.

[211] Peixiang Zhao, Jiawei Han, and Yizhou Sun. P-rank: a comprehensive structural similarity measure over information networks. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 553–562, New York, NY, USA, 2009. ACM.

[212] Xiaoqing Zheng, Jiangtao Feng, Mengxiao Lin, and Wenqiang Zhang. Context-specific and multi-prototype character representations. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 3007–3013. AAAI Press, 2016.

[213] Zhi Zhong and Hwee Tou Ng. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, ACLDemos '10, pages 78–83, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[214] Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Stat. Anal. Data Min.*, 5(5):363–387, October 2012.

[215] Guang Y. Zou. Toward using confidence intervals to compare correlations. *Psychological methods*, 12(4):399–413, December 2007.

Appendix A

An Introduction To The WikiSim Architecture

We publish a semantic analysis system, "WikiSim" [1], that includes all the code, tools and data to replicate our experiments or to incorporate our solution in similar systems. While the main purpose of WikiSim was providing a tool for *concept embedding* and wikification, the final product is also a framework that integrates many other tools and resources. We start from the main features of the system from the user perspective and proceed to explain a summary of the architecture of the system. The code is hosted on Github under MIT license.

A.1 API features

Wikisim provides different levels of information about the structure and the content of Wikipedia, including:

A.2 Concept Embedding

Given a concept, returns an embedding using various methods:

1. *rvsPagerank*: cf. Section 3.4.5
2. *word2vec*: cf. Section 3.8.1
3. *Node Embedding*: cf. Section 3.6.1

A.2.1 Semantic Similarity

Calculating the similarity using different methods:

1. *Graph Overlap* or *bibliometric measures*, including *cocitation* (cf. Eq. 2.15), *coupling* (cf. Eq. 2.16), and *amsler* (cf. Eq. 2.17)

[1] `https://github.com/asajadi/wikisim`

2. Normalized Graph Edit Distance (NGED): cf. Eq. 3.19

3. Wikipedia Link Measure (WLM) (a.k.a. Wikipedia Miner): cf. Eq. 2.10

4. Normalized Google Distance (NGD): cf. Eq. 3.22

5. *Word2vec*, cf. Section 3.8.1

6. *Node Embedding*: Global node embedding of the graph, cf. Section 3.6.1

7. *rvsPagerank*: $cosine$ similarity between $rvsPagerank$ embeddings, cf. Section 3.4.5

A.2.2 Entity Linking API

1. Given a text, splits it into mentions, i.e, substrings that possibly refer to some Wikipedia concepts.

2. Returns possible candidates along with various measures, such as *popularity*, *semantic coherence* (Section 4.5) and *textual context similarity*.

3. Disambiguate the mentions using the machine learning model, trained on different features.

A.2.3 Structure API

Given a node, it can answer various questions such as:

1. The *title*, *id*, whether it is a *concept* or *directory*, *main* or *redirect* concept

2. *Incoming* or *outgoing* links

3. The directed neighbourhood graphs (using several optimization and indexing techniques)

4. *Concept-category* and *subcategory-category* relations.

A.2.4 Text API

Regarding the text of *Wikipedia*, we are having multiple instances of Solr [2] cores, indexing different features of Wikipedia. Some examples of the possible queries it can answer are:

1. Given a concept, return the opening text, or full text associated with it.
2. Given a term, return all the concepts (titles) of the pages in which it appears.
3. Given a term, return all windows (of size 20) of contexts in which it appears.
4. Given a term, return *true* if it is ever used as an anchor text, along with the entities it is linked to.

A.3 REST API

We provide a limited API through REST: embedding, semantic relatedness (single or batch mode) and wikification. Fig. A.1 shows a snapshot of the system, with embeddings visualized as word clouds.

A.4 Architecture

We use different tools and technologies to build the whole system. Aside from the third party libraries, almost all the code is written in python, presented as Jupyter Notebooks [3]. A high level illustration of the main modules of the system is shown in Fig. A.2.

A.4.1 Data Importer

This module is a collection of codes written by us and also, third party tools and libraries. Data importing was a very challenging part of the development and also, the hardest part to maintain. Some of the main submodules are:

1. Database Normalizer: It is a piece of code written by us in Java that parses the database dumps of Wikipedia and normalizes the data. Some of the more important preprocessing steps performed on the data are: removing *none concept* and *none*

[2] `http://lucene.apache.org/solr/`
[3] `http://jupyter.org/`

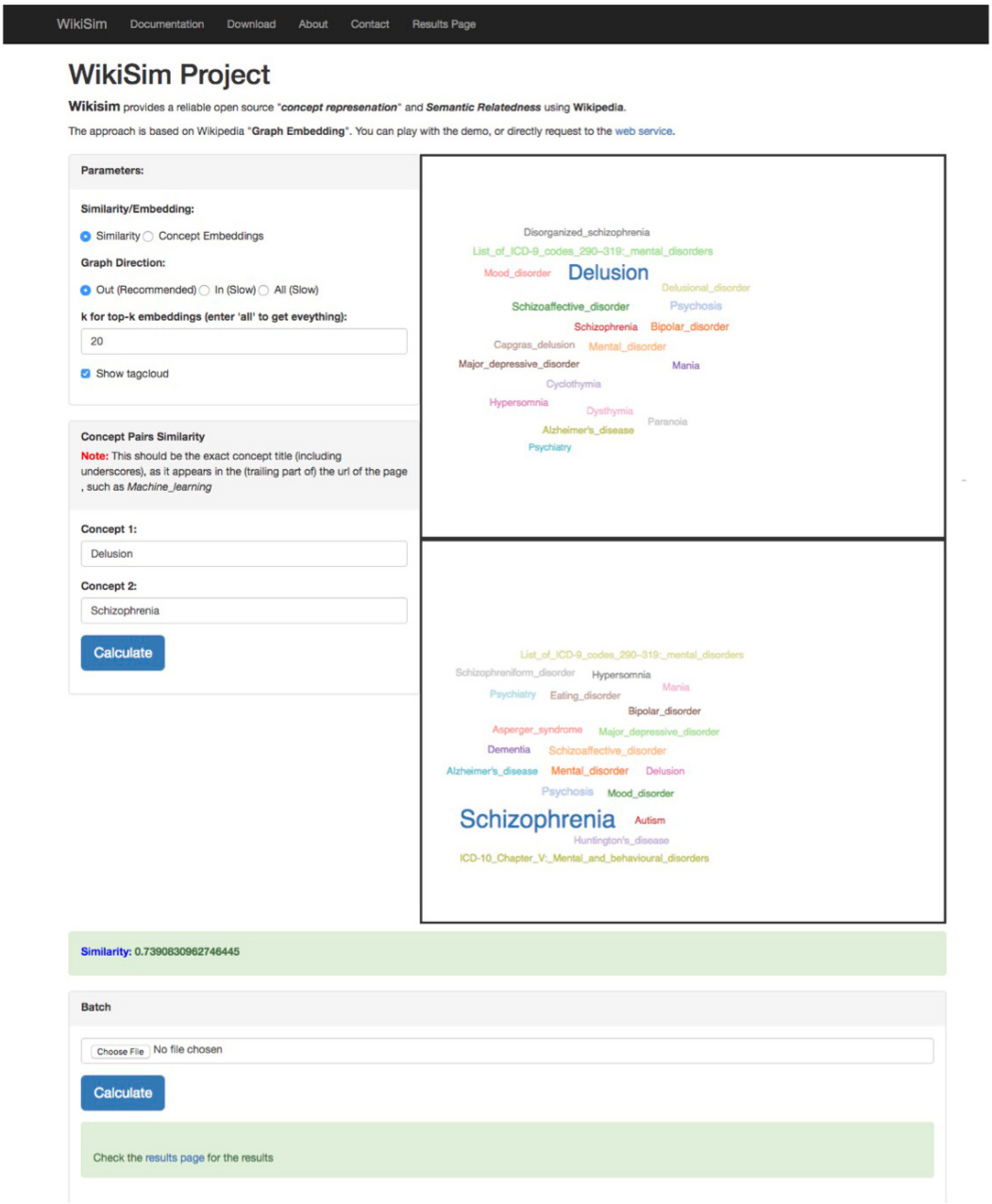


Figure A.1: A snapshot of the system: visualized embeddings for two words, *Schizophrenia* and *Delusion*, and their relatedness.

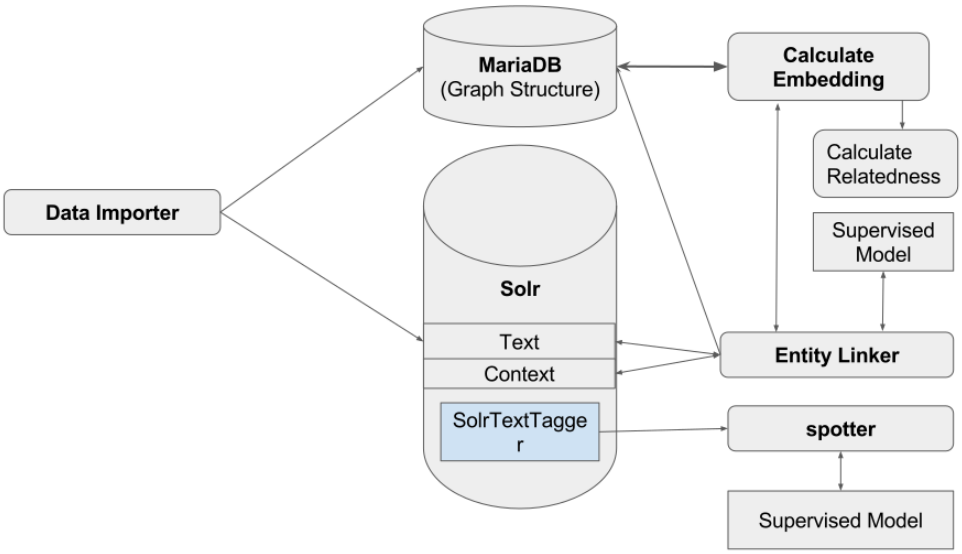


Figure A.2: A modular illustration of Wikisim architecture: We use one sided arrows to denote *is-used-by* relation.

category entries, removing *dead redirects*, calculating *synonym sets*, normalizing the links by removing *redirects* and *dead links*, optimizing for fast neighbourhood retrieval, etc.

2. WikiExtractor [4]: A parser for Wikipedia Text

3. Text importer: These are a set of scripts to extract the links, anchor texts, context for each anchor text and other information from the result of WikiExtractor.

A.4.2 MariaDB

We use MariaDB [5] to manage the graph, it stores all the information in multiple tables, with many different indices to speedup different queries. Some of the more important tables are:

1. Concept information.

2. Redirection

3. Graph

[4] https://github.com/attardi/wikiextractor
[5] https://mariadb.org/

4. Categories

5. Pre-calculated embeddings

A.4.3 Solr

We heavily rely on Solr [6] to store different aspects of the Wikipedia text, such as:

1. The opening text associated with each page

2. The text associated with each page

3. The list of the textual contexts (of size 20) in which a concept appears.

A.4.4 SolrTextTagger

We use SolrTextTagger [7] to spot the mentions. SolrTextTagger is an add on to Solr, relying on the internal implementation of a *Finite State Transducer* (FST) to spot the mentions in the text. SolrTextTagger can potentially find all mentions. It also can be setup for various strategies, such as greedily spotting the longest mentions to avoid overlap.

A.4.5 Calculating Embedding And Semantic Relatedness

Given two concepts, these two modules calculate the embedding and semantic relatedness respectively.

A.4.6 Spotter

The spotter is responsible for mention detection. We have indexed an exhaustive list of all anchors in Solr and use SolrTextTagger for doing the initial spotting. This initial result is given to a trained model that can classify mentions as *true* or *false* positives.

A.4.7 Linker

This module takes a mention input from the spotter, then retrieves various features from Solr, the database, and internal modules (such as coherence measure). It then relies on a trained model (a *rank learner*) to make the final decision.

[6] `http://lucene.apache.org/solr/`
[7] `https://github.com/OpenSextant/SolrTextTagger`

A.5 Conclusion

We discussed a brief introduction to the open-source framework resulted from this study. The implementation uses several data management and text mining tools to provide a wide range of API. The provided API and web service makes it simple to incorporate the system in other applications.

Appendix B

Fast Pagerank Implementation

While there exist several implementations of Pagerank, we were not able to find one that suites the needs of our system, that is, an implementation that works with the native sparse-matrix representation of the graphs. Considering the frequency of using this feature, back and forth translation of the graphs into the representations acceptable by these third-party libraries is computationally expensive. We implemented our own Pagerank using the algorithm explained in [140], which we refer to as "Moler Pagerank". We implemented two different versions of this algorithm: an exact solution based on solving a *sparse linear system* (source code B.1) and an approximation using *power method* (source code B.2). We also modified the algorithm to calculate *Personalized Pagerank* as well. In this case, there is another extra vector which represents the *teleporting* vector preference, i.e, an initial probability distribution over the nodes.

The main advantage of these implementations relies in maintaining the sparsity of the matrices at every stage of the computation, and taking advantage of built-in sparse matrix operations. The performance of the code depends on how every instruction is implemented, therefore we quote the Python code instead of a high level algorithm.

Our benchmarks (Fig. B.1 and fig. B.2) show that our implementations are significantly faster than the popular implementations of the networkx[1] library. In fact, it is the fastest Python implementation to the best of our knowledge.

[1] `http://networkx.readthedocs.io/en/networkx-1.10/`

```
1
2  def moler_pagerank_sparse(G, p, pv):
3      """
4      Args:
5          G: a csr graph.
6          p: teleporting probability
7          pv: vector of probability distrib. over the nodes
8      Returns:
9          Pagerank scores for the nodes
10     """
11     # In Moler's algorithm, G[i,j] represents the existences of an edge
12   # from node j to i, while we have assumed the opposite!
13     G = G.T
14     n,_ = G.shape
15     c = sp.asarray(G.sum(axis=0)).reshape(-1)
16     k = c.nonzero()[0]
17     D = sprs.csr_matrix((1/c[k],(k,k)), shape=(n,n))
18     pv = pv.reshape(n,1)
19     e = n*pv
20     I = sprs.eye(n)
21     x = sprs.linalg.spsolve((I - p*G.dot(D)), e);
22     x = x/x.sum()
23     return x
```

Listing B.1: Calculating Pagerank by solving a *sparse linear system*

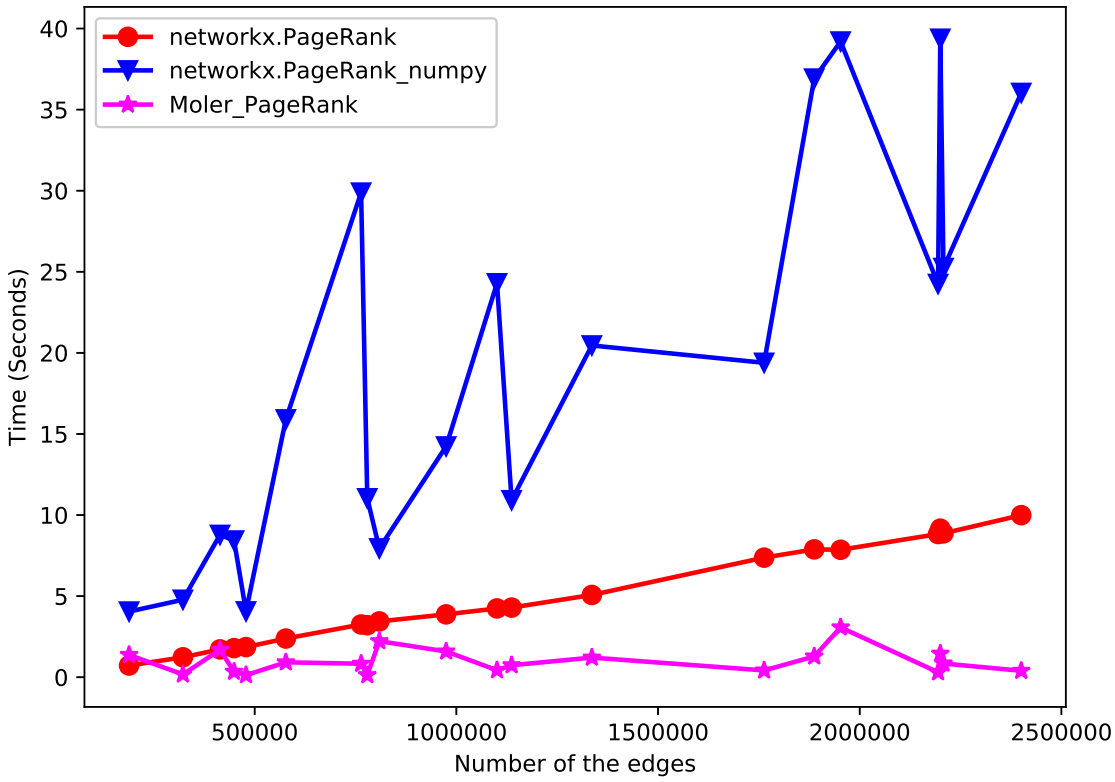


Figure B.1: Comparing our implementation (*Moler Pagerank*) of exact solution for Pagerank with *networkx*'s implementations. We use *linear-equation solver*, while *networkx* uses a merely python implementation in *networkx.pagerank* and *eigenvalues* based solution in *networkx.pagerank_numpy*

```
def moler_pagerank_sparse_power(G, p, tol, pv):
    """
    Args:
        G: a csr graph.
        p: teleporting probability
        tol: threshold for convergence
        pv: vector of probability distrib. over the nodes
    Returns:
        Pagerank Scores for the nodes
    """
    # In Moler's algorithm, G[i,j] represents the existences of an edge
    # from node j to i, while we have assumed the opposite!
    G = G.T
    n,_ = G.shape
    c = sp.asarray(G.sum(axis=0)).reshape(-1)
    k = c.nonzero()[0]
    D = sprs.csr_matrix((1/c[k],(k,k)),shape=(n, n))
    pv = pv.reshape(n,1)
    e = (pv/pv.sum())*n
    z = (((1-p)*(c!=0) + (c==0))/n)[sp.newaxis,:]
    G = p*G.dot(D)
    x = e/n
    oldx = sp.zeros((n,1));

    while sp.linalg.norm(x-oldx) > tol:
        oldx = x
        x = G.dot(x) + e.dot(z.dot(x))
    x = x/sum(x)
    return x.reshape(-1)

```

Listing B.2: Calculating Pagerank using *power method*

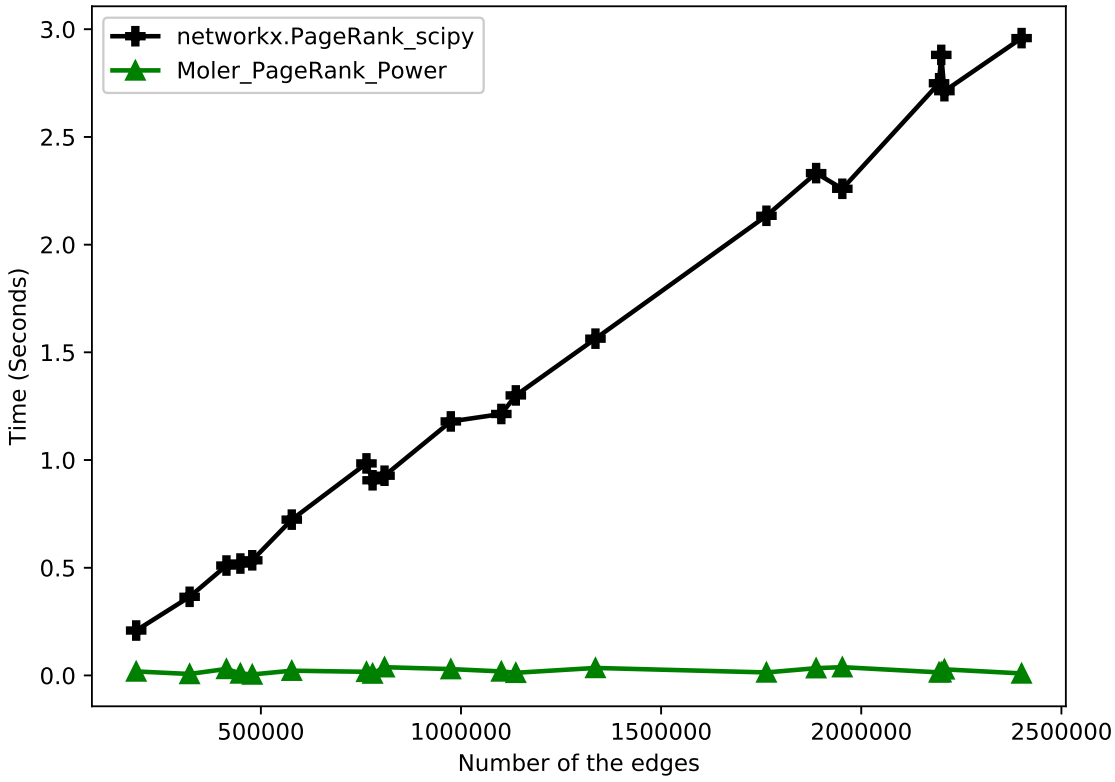


Figure B.2: Comparing our implementation (*Moler Pagerank*) of approximate solution for Pagerank with the *networkx*'s implementation. Both implementations are based on Power Method

Appendix C

Copyright Permission

This appendix includes the copyright forms for our publications in:

1. 27th Canadian Conference on Artificial Intelligence (AI 2014) [191]
2. 27th Computational Linguistics and Intelligent Text Processing: 16th International Conference, CICLing 2015 [181]
3. 22nd International Conference on Applications of Natural Language to Information Systems, NLDB 2017 [180]

Consent to Publish

Lecture Notes in Computer Science

Springer

Title of the Book or Conference Name: 27th Canadian Conference on Artificial Intelligence (AI 201

Volume Editor(s): Marina Sokolova and Peter van Beek

Title of the Contribution: Graph-Based Domain-Specific Relatedness from Wikipedia

Author(s) Name(s): Armin Sajadi

Corresponding Author's Name, Address, Affiliation and Email:

Dalhousie University, Halifax, NS B3H 4R2, Canada

sajadi@cs.dal.ca

When Author is more than one person the expression "Author" as used in this agreement will apply collectively unless otherwise indicated.

§ 1 Rights Granted

Author hereby grants and assigns to Springer International Publishing AG, Cham (hereinafter called Springer) the exclusive, sole, permanent, world-wide, transferable, sub-licensable and unlimited right to reproduce, publish, distribute, transmit, make available or otherwise communicate to the public, translate, publicly perform, archive, store, lease or lend and sell the Contribution or parts thereof individually or together with other works in any language, in all revisions and versions (including soft cover, book club and collected editions, anthologies, advance printing, reprints or print to order, microfilm editions, audiograms and videograms), in all forms and media of expression including in electronic form (including offline and online use, push or pull technologies, use in databases and networks for display, print and storing on any and all stationary or portable end-user devices, e.g. text readers, audio, video or interactive devices, and for use in multimedia or interactive versions as well as for the display or transmission of the Contribution or parts thereof in data networks or seach engines), in whole, in part or in abridged form, in each case as now known or developed in the future, including the right to grant further time-limited or permanent rights. For the purposes of use in electronic forms, Springer may adjust the Contribution to the respective form of use and include links or otherwise combine it with other works. For the avoidance of doubt, Springer has the right to permit others to use individual illustrations and may use the Contribution for advertising purposes.

The copyright of the Contribution will be held in the name of Springer. Springer may take, either in its own name or in that of copyright holder, any necessary steps to protect these rights against infringement by third parties. It will have the copyright notice inserted into all editions of the Contribution according to the provisions of the Universal Copyright Convention (UCC) and dutifully take care of all formalities in this connection in the name of the copyright holder.

§ 2 Regulations for Authors under Special Copyright Law

The parties acknowledge that there may be no basis for claim of copyright in the United States to a Contribution prepared by an officer or employee of the United States government as part of that person's official duties. If the Contribution was performed under a United States government contract, but Author is not a United States government employee, Springer grants the United States government royalty-free permission to reproduce all or part of the Contribution and to authorize others to do so for United States government purposes.

If the Contribution was prepared or published by or under the direction or control of Her Majesty (i.e., the constitutional monarch of the Commonwealth realm) or any Crown government department, the copyright in the Contribution shall, subject to any agreement with Author, belong to Her Majesty.

If the Contribution was created by an employee of the European Union or the European Atomic Energy Community (EU/Euratom) in the performance of their duties, the regulation 31/EEC, 11/EAEC (Staff Regulations) applies, and copyright in the Contribution shall, subject to the Publication Framework Agreement (EC Plug), belong to the European Union or the European Atomic Energy Community.

If Author is an officer or employee of the United States government, of the Crown, or of EU/Euratom, reference will be made to this status on the signature page.

§ 3 Rights Retained by Author

Author retains, in addition to uses permitted by law, the right to communicate the content of the Contribution to other scientists, to share the Contribution with them in manuscript form, to perform or present the Contribution or to use the content for non-commercial internal and educational purposes, provided the Springer publication is mentioned as the

original source of publication in any printed or electronic materials. Author retains the right to republish the Contribution in any collection consisting solely of Author's own works without charge subject to ensuring that the publication by Springer is properly credited and that the relevant copyright notice is repeated verbatim.
Author may self-archive an author-created version of his/her Contribution on his/her own website and/or the repository of Author's department or faculty. Author may also deposit this version on his/her funder's or funder's designated repository at the funder's request or as a result of a legal obligation. He/she may not use the publisher's PDF version, which is posted on SpringerLink and other Springer websites, for the purpose of self-archiving or deposit. Furthermore, Author may only post his/her own version, provided acknowledgment is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be provided by inserting the DOI number of the article in the following sentence: "The final publication is available at Springer via http://dx.doi.org/[insert DOI]". The DOI (Digital Object Identifier) can be found at the bottom of the first page of the published paper.
Prior versions of the Contribution published on non-commercial pre-print servers like ArXiv/CoRR and HAL can remain on these servers and/or can be updated with Author's accepted version. The final published version (in pdf or html/xml format) cannot be used for this purpose. Acknowledgment needs to be given to the final publication and a link must be inserted to the published Contribution on Springer's website, by inserting the DOI number of the article in the following sentence: "The final publication is available at Springer via http://dx.doi.org/[insert DOI]".
Author retains the right to use his/her Contribution for his/her further scientific career by including the final published paper in his/her dissertation or doctoral thesis provided acknowledgment is given to the original source of publication. Author also retains the right to use, without having to pay a fee and without having to inform the publisher, parts of the Contribution (e.g. illustrations) for inclusion in future work, and to publish a substantially revised version (at least 30% new content) elsewhere, provided that the original Springer Contribution is properly cited.

§ 4 Warranties

Author warrants that the Contribution is original except for such excerpts from copyrighted works (including illustrations, tables, animations and text quotations) as may be included with the permission of the copyright holder thereof, in which case(s) Author is required to obtain written permission to the extent necessary and to indicate the precise sources of the excerpts in the manuscript. Author is also requested to store the signed permission forms and to make them available to Springer if required.
Author warrants that he/she is entitled to grant the rights in accordance with Clause 1 "Rights Granted", that he/she has not assigned such rights to third parties, that the Contribution has not heretofore been published in whole or in part, that the Contribution contains no libelous statements and does not infringe on any copyright, trademark, patent, statutory right or proprietary right of others, including rights obtained through licenses; and that Author will indemnify Springer against any costs, expenses or damages for which Springer may become liable as a result of any breach of this warranty.

§ 5 Delivery of the Work and Publication

Author agrees to deliver to the responsible Volume Editor (for conferences, usually one of the Program Chairs), on a date to be agreed upon, the manuscript created according to the Springer Instructions for Authors. Springer will undertake the reproduction and distribution of the Contribution at its own expense and risk. After submission of the Consent to Publish form Signed by the Corresponding Author, changes of authorship, or in the order of the authors listed, will not be accepted by Springer.

§ 6 Author's Discount

Author is entitled to purchase for his/her personal use (directly from Springer) the Work or other books published by Springer at a discount of 33 1/3% off the list price as long as there is a contractual arrangement between Author and Springer and subject to applicable book price regulation. Resale of such copies or of free copies is not permitted.

§ 7 Governing Law and Jurisdiction

This agreement shall be governed by, and shall be construed in accordance with, the laws of Switzerland. The courts of Zug, Switzerland shall have the exclusive jurisdiction.

Corresponding Author signs for and accepts responsibility for releasing this material on behalf of any and all Co-authors.

Signature of Corresponding Author: **Date:** 2014-03-02

. . . . . .

☐ I'm an employee of the US Government and transfer the rights to the extent transferable (Title 17 §105 U.S.C. applies)

☐ I'm an employee of the Crown and copyright on the Contribution belongs to Her Majesty

☐ I'm an employee of the EU or Euratom and copyright on the Contribution belongs to EU or Euratom

Consent to Publish

Lecture Notes in Computer Science

Springer

Title of the Book or Conference Name: COMPUTATIONAL LINGUISTICS AND INTELLIGENT TEXT PROCESSING

Volume Editor(s): ALEXANDER GELBUKH

Title of the Contribution: Domain-specific Semantic Relatedness from Wikipedia Structure: A Case Study in Biomedical Text

Author(s) Name(s): Armin Sajadi, Evangelos E. Milios, and Vlado Keselj

Corresponding Author's Name, Address, Affiliation and Email: Armin Sajadi
Dalhousie University, Faculty of Computer Science, Halifax, NS, Canada B3H 4R2
sajadi@cs.dal.ca

When Author is more than one person the expression "Author" as used in this agreement will apply collectively unless otherwise indicated.

§ 1 Rights Granted

Author hereby grants and assigns to Springer International Publishing AG, Cham (hereinafter called Springer) the exclusive, sole, permanent, world-wide, transferable, sub-licensable and unlimited right to reproduce, publish, distribute, transmit, make available or otherwise communicate to the public, translate, publicly perform, archive, store, lease or lend and sell the Contribution or parts thereof individually or together with other works in any language, in all revisions and versions (including soft cover, book club and collected editions, anthologies, advance printing, reprints or print to order, microfilm editions, audiograms and videograms), in all forms and media of expression including in electronic form (including offline and online use, push or pull technologies, use in databases and networks for display, print and storing on any and all stationary or portable end-user devices, e.g. text readers, audio, video or interactive devices, and for use in multimedia or interactive versions as well as for the display or transmission of the Contribution or parts thereof in data networks or seach engines), in whole, in part or in abridged form, in each case as now known or developed in the future, including the right to grant further time-limited or permanent rights. For the purposes of use in electronic forms, Springer may adjust the Contribution to the respective form of use and include links or otherwise combine it with other works. For the avoidance of doubt, Springer has the right to permit others to use individual illustrations and may use the Contribution for advertising purposes.
The copyright of the Contribution will be held in the name of Springer. Springer may take, either in its own name or in that of copyright holder, any necessary steps to protect these rights against infringement by third parties. It will have the copyright notice inserted into all editions of the Contribution according to the provisions of the Universal Copyright Convention (UCC) and dutifully take care of all formalities in this connection in the name of the copyright holder.

§ 2 Regulations for Authors under Special Copyright Law

The parties acknowledge that there may be no basis for claim of copyright in the United States to a Contribution prepared by an officer or employee of the United States government as part of that person's official duties. If the Contribution was performed under a United States government contract, but Author is not a United States government employee, Springer grants the United States government royalty-free permission to reproduce all or part of the Contribution and to authorize others to do so for United States government purposes.
If the Contribution was prepared or published by or under the direction or control of Her Majesty (i.e., the constitutional monarch of the Commonwealth realm) or any Crown government department, the copyright in the Contribution shall, subject to any agreement with Author, belong to Her Majesty.
If the Contribution was created by an employee of the European Union or the European Atomic Energy Community (EU/Euratom) in the performance of their duties, the regulation 31/EEC, 11/EAEC (Staff Regulations) applies, and copyright in the Contribution shall, subject to the Publication Framework Agreement (EC Plug), belong to the European Union or the European Atomic Energy Community.
If Author is an officer or employee of the United States government, of the Crown, or of EU/Euratom, reference will be made to this status on the signature page.

§ 3 Rights Retained by Author

Author retains, in addition to uses permitted by law, the right to communicate the content of the Contribution to other scientists, to share the Contribution with them in manuscript form, to perform or present the Contribution or to use the content for non-commercial internal and educational purposes, provided the Springer publication is mentioned as the

original source of publication in any printed or electronic materials. Author retains the right to republish the Contribution in any collection consisting solely of Author's own works without charge subject to ensuring that the publication by Springer is properly credited and that the relevant copyright notice is repeated verbatim.
Author may self-archive an author-created version of his/her Contribution on his/her own website and/or the repository of Author's department or faculty. Author may also deposit this version on his/her funder's or funder's designated repository at the funder's request or as a result of a legal obligation. He/she may not use the publisher's PDF version, which is posted on SpringerLink and other Springer websites, for the purpose of self-archiving or deposit. Furthermore, Author may only post his/her own version, provided acknowledgment is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be provided by inserting the DOI number of the article in the following sentence: "The final publication is available at Springer via http://dx.doi.org/[insert DOI]". The DOI (Digital Object Identifier) can be found at the bottom of the first page of the published paper.
Prior versions of the Contribution published on non-commercial pre-print servers like ArXiv/CoRR and HAL can remain on these servers and/or can be updated with Author's accepted version. The final published version (in pdf or html/xml format) cannot be used for this purpose. Acknowledgment needs to be given to the final publication and a link must be inserted to the published Contribution on Springer's website, by inserting the DOI number of the article in the following sentence: "The final publication is available at Springer via http://dx.doi.org/[insert DOI]".
Author retains the right to use his/her Contribution for his/her further scientific career by including the final published paper in his/her dissertation or doctoral thesis provided acknowledgment is given to the original source of publication. Author also retains the right to use, without having to pay a fee and without having to inform the publisher, parts of the Contribution (e.g. illustrations) for inclusion in future work, and to publish a substantially revised version (at least 30% new content) elsewhere, provided that the original Springer Contribution is properly cited.

§ 4 Warranties

Author warrants that the Contribution is original except for such excerpts from copyrighted works (including illustrations, tables, animations and text quotations) as may be included with the permission of the copyright holder thereof, in which case(s) Author is required to obtain written permission to the extent necessary and to indicate the precise sources of the excerpts in the manuscript. Author is also requested to store the signed permission forms and to make them available to Springer if required.
Author warrants that he/she is entitled to grant the rights in accordance with Clause 1 "Rights Granted", that he/she has not assigned such rights to third parties, that the Contribution has not heretofore been published in whole or in part, that the Contribution contains no libelous statements and does not infringe on any copyright, trademark, patent, statutory right or proprietary right of others, including rights obtained through licenses; and that Author will indemnify Springer against any costs, expenses or damages for which Springer may become liable as a result of any breach of this warranty.

§ 5 Delivery of the Work and Publication

Author agrees to deliver to the responsible Volume Editor (for conferences, usually one of the Program Chairs), on a date to be agreed upon, the manuscript created according to the Springer Instructions for Authors. Springer will undertake the reproduction and distribution of the Contribution at its own expense and risk. After submission of the Consent to Publish form Signed by the Corresponding Author, changes of authorship, or in the order of the authors listed, will not be accepted by Springer.

§ 6 Author's Discount

Author is entitled to purchase for his/her personal use (directly from Springer) the Work or other books published by Springer at a discount of 33 1/3% off the list price as long as there is a contractual arrangement between Author and Springer and subject to applicable book price regulation. Resale of such copies or of free copies is not permitted.

§ 7 Governing Law and Jurisdiction

This agreement shall be governed by, and shall be construed in accordance with, the laws of Switzerland. The courts of Zug, Switzerland shall have the exclusive jurisdiction.

Corresponding Author signs for and accepts responsibility for releasing this material on behalf of any and all Co-authors.

Signature of Corresponding Author: **Date:** March 3, 2015

. . . . . .

☐ I'm an employee of the US Government and transfer the rights to the extent transferable (Title 17 §105 U.S.C. applies)

☐ I'm an employee of the Crown and copyright on the Contribution belongs to Her Majesty

☐ I'm an employee of the EU or Euratom and copyright on the Contribution belongs to EU or Euratom

Consent to Publish

Lecture Notes in Computer Science

Springer

Title of the Book or Conference Name: Natural Language Processing and Information Systems

Volume Editor(s): Flavius Frasincar, Ashwin Ittoo, Le Minh Nguyen, and Elisabeth Métais

Title of the Contribution: Vector Space Representation of Concepts Using Wikipedia Graph Structure

Author(s) Name(s): Armin Sajadi, Evangelos E. Milios, and Vlado Keselj

Corresponding Author's Name, Address, Affiliation and Email:

Dalhousie University, Faculty of Computer Science, Halifax, NS, Canada B3H 4R2

sajadi@cs.dal.ca

When Author is more than one person the expression "Author" as used in this agreement will apply collectively unless otherwise indicated.

§ 1 Rights Granted

Author hereby grants and assigns to Springer International Publishing AG, Cham (hereinafter called Springer) the exclusive, sole, permanent, world-wide, transferable, sub-licensable and unlimited right to reproduce, publish, distribute, transmit, make available or otherwise communicate to the public, translate, publicly perform, archive, store, lease or lend and sell the Contribution or parts thereof individually or together with other works in any language, in all revisions and versions (including soft cover, book club and collected editions, anthologies, advance printing, reprints or print to order, microfilm editions, audiograms and videograms), in all forms and media of expression including in electronic form (including offline and online use, push or pull technologies, use in databases and networks for display, print and storing on any and all stationary or portable end-user devices, e.g. text readers, audio, video or interactive devices, and for use in multimedia or interactive versions as well as for the display or transmission of the Contribution or parts thereof in data networks or seach engines), in whole, in part or in abridged form, in each case as now known or developed in the future, including the right to grant further time-limited or permanent rights. For the purposes of use in electronic forms, Springer may adjust the Contribution to the respective form of use and include links or otherwise combine it with other works. For the avoidance of doubt, Springer has the right to permit others to use individual illustrations and may use the Contribution for advertising purposes.

The copyright of the Contribution will be held in the name of Springer. Springer may take, either in its own name or in that of copyright holder, any necessary steps to protect these rights against infringement by third parties. It will have the copyright notice inserted into all editions of the Contribution according to the provisions of the Universal Copyright Convention (UCC) and dutifully take care of all formalities in this connection in the name of the copyright holder.

§ 2 Regulations for Authors under Special Copyright Law

The parties acknowledge that there may be no basis for claim of copyright in the United States to a Contribution prepared by an officer or employee of the United States government as part of that person's official duties. If the Contribution was performed under a United States government contract, but Author is not a United States government employee, Springer grants the United States government royalty-free permission to reproduce all or part of the Contribution and to authorize others to do so for United States government purposes.

If the Contribution was prepared or published by or under the direction or control of Her Majesty (i.e., the constitutional monarch of the Commonwealth realm) or any Crown government department, the copyright in the Contribution shall, subject to any agreement with Author, belong to Her Majesty.

If the Contribution was created by an employee of the European Union or the European Atomic Energy Community (EU/Euratom) in the performance of their duties, the regulation 31/EEC, 11/EAEC (Staff Regulations) applies, and copyright in the Contribution shall, subject to the Publication Framework Agreement (EC Plug), belong to the European Union or the European Atomic Energy Community.

If Author is an officer or employee of the United States government, of the Crown, or of EU/Euratom, reference will be made to this status on the signature page.

§ 3 Rights Retained by Author

Author retains, in addition to uses permitted by law, the right to communicate the content of the Contribution to other scientists, to share the Contribution with them in manuscript form, to perform or present the Contribution or to use the content for non-commercial internal and educational purposes, provided the Springer publication is mentioned as the

original source of publication in any printed or electronic materials. Author retains the right to republish the Contribution in any collection consisting solely of Author's own works without charge subject to ensuring that the publication by Springer is properly credited and that the relevant copyright notice is repeated verbatim.
Author may self-archive an author-created version of his/her Contribution on his/her own website and/or the repository of Author's department or faculty. Author may also deposit this version on his/her funder's or funder's designated repository at the funder's request or as a result of a legal obligation. He/she may not use the publisher's PDF version, which is posted on SpringerLink and other Springer websites, for the purpose of self-archiving or deposit. Furthermore, Author may only post his/her own version, provided acknowledgment is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be provided by inserting the DOI number of the article in the following sentence: "The final publication is available at Springer via http://dx.doi.org/[insert DOI]". The DOI (Digital Object Identifier) can be found at the bottom of the first page of the published paper.
Prior versions of the Contribution published on non-commercial pre-print servers like ArXiv/CoRR and HAL can remain on these servers and/or can be updated with Author's accepted version. The final published version (in pdf or html/xml format) cannot be used for this purpose. Acknowledgment needs to be given to the final publication and a link must be inserted to the published Contribution on Springer's website, by inserting the DOI number of the article in the following sentence: "The final publication is available at Springer via http://dx.doi.org/[insert DOI]".
Author retains the right to use his/her Contribution for his/her further scientific career by including the final published paper in his/her dissertation or doctoral thesis provided acknowledgment is given to the original source of publication. Author also retains the right to use, without having to pay a fee and without having to inform the publisher, parts of the Contribution (e.g. illustrations) for inclusion in future work, and to publish a substantially revised version (at least 30% new content) elsewhere, provided that the original Springer Contribution is properly cited.

§ 4 Warranties

Author warrants that the Contribution is original except for such excerpts from copyrighted works (including illustrations, tables, animations and text quotations) as may be included with the permission of the copyright holder thereof, in which case(s) Author is required to obtain written permission to the extent necessary and to indicate the precise sources of the excerpts in the manuscript. Author is also requested to store the signed permission forms and to make them available to Springer if required.
Author warrants that he/she is entitled to grant the rights in accordance with Clause 1 "Rights Granted", that he/she has not assigned such rights to third parties, that the Contribution has not heretofore been published in whole or in part, that the Contribution contains no libelous statements and does not infringe on any copyright, trademark, patent, statutory right or proprietary right of others, including rights obtained through licenses; and that Author will indemnify Springer against any costs, expenses or damages for which Springer may become liable as a result of any breach of this warranty.

§ 5 Delivery of the Work and Publication

Author agrees to deliver to the responsible Volume Editor (for conferences, usually one of the Program Chairs), on a date to be agreed upon, the manuscript created according to the Springer Instructions for Authors. Springer will undertake the reproduction and distribution of the Contribution at its own expense and risk. After submission of the Consent to Publish form Signed by the Corresponding Author, changes of authorship, or in the order of the authors listed, will not be accepted by Springer.

§ 6 Author's Discount

Author is entitled to purchase for his/her personal use (directly from Springer) the Work or other books published by Springer at a discount of 40% off the list price as long as there is a contractual arrangement between Author and Springer and subject to applicable book price regulation. Resale of such copies or of free copies is not permitted.

§ 7 Governing Law and Jurisdiction

This agreement shall be governed by, and shall be construed in accordance with, the laws of Switzerland. The courts of Zug, Switzerland shall have the exclusive jurisdiction.

Corresponding Author signs for and accepts responsibility for releasing this material on behalf of any and all Co-authors.

Signature of Corresponding Author: **Date:** 26/03/2017

. . . . . .

I'm an employee of the US Government and transfer the rights to the extent transferable
(Title 17 §105 U.S.C. applies)

I'm an employee of the Crown and copyright on the Contribution belongs to Her Majesty

I'm an employee of the EU or Euratom and copyright on the Contribution belongs to EU or Euratom