A Knowledge-based Approach for Semantic Similarity, Relatedness, and Word Sense Disambiguation using WordNet

Final Report for Engi-9900-GA Ph.D. Dissertation

Author: Mohannad Adel Al-Mousa (malmous@lakeheadu.ca)

Supervisor: Dr. Rachid Benlamri, Dr. Richard Khoury

(rbenlamr@lakeheadu.ca, richard.khoury@ift.ulaval.ca)

September 2020

Version: 2.0 (Release)

This thesis was submitted as partial fulfilment of a PhD degree in

Electrical and Computer Engineering - Software Engineering

(Engi-9900-GA)

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: Diana Inkpen
Professor, School of Electrical Engineering and Computer Science,
University of Ottawa, Canada

Internal Members: Abdulsalam Yassine
Assistant Professor, Dept. of Software Engineering,
Lakehead University, Canada

Salama Ikki
Associate Professor, Dept. of Electrical Engineering,
Lakehead University, Canada

Supervisor: Rachid Benlamri
Professor, Dept. of Software Engineering,
Lakehead University, Canada

Co-Supervisor: Richard Khoury
Associate Professor, Dept. of Computer Science & Software Engineering,
Université Laval, Canada

Declaration of Originality

I confirm that:

- This submission is my own work, except where clearly indicated.
- In submitting this work I understand and agree to abide by the University's regulations governing these issues.

Name ..... Mohannad Adel Al-Mousa .....

Date ..... October 26th, 2020 .....

Consent to Share this Work

By including my name below, I hereby agree to this dissertation being made available to other students and academic staff of the Lakehead University Software Engineering Department.

Name ..... Mohannad Adel Al-Mousa .....

Date ..... October 26th, 2020 .....

Dedication

I dedicate my dissertation work to my beloved parents Adel Al-Mousa and Widad Atarah, for their unconditional love and care, education, and motivating me to pursue my Ph.D.

I dedicate my dissertation work to my brothers Amer, Ahmad, and Abdallah and lovely sister Dina, who have always been by my side with their unconditional support.

I dedicate my dissertation work to all friends and family members, who have been a great support during my research.

Last but not least, I dedicate my dissertation work to my beloved wife Fatima Bani-Hani, for years of patience, support, and encouragement throughout my life and specifically the Ph.D. years, as she has gone above and beyond to support our family emotionally and financially putting the family before herself. To my lovely children Jenna, Abrar, Alia, Mohammad, and Maryam, for their continuous prayers, understanding, and tolerating their busy father when I couldn't join their joyful moments.

Acknowledgements

First and foremost, praises and thanks to Allah, the Almighty, for showering me with myriad blessings thought my life, especially during my Ph.D. journey.

I would like to express my deep and sincere gratitude to my research supervisors, Dr. Rachid Benlamri and Dr. Richard Khoury, for providing invaluable guidance throughout this research. Their vision, sincerity, and motivation have deeply inspired me. They have taught me the methodology to carry out research and present it as clearly as possible.

I would like to thanks my friends and research colleagues, Eduardo Reis, Seyednima Khezr, Dr. Ayman Alahmar, Md Moniruzzaman, and Andrew Tittaferrante for being there whenever I needed them.

I would like to express my special thanks to Mr. Peter Haslam for professionally proofreading my dissertation on short notice despite his busy schedule. I also want to thank my dear friends, Farhan Yousaf, Ibrahim Al-Hurani, Malik Alsmadi, Omar Qananeweh, and Abdellah Al-Mousa, who helped proofread various sections of my dissertation.

I would like to thank all faculty members of the Department of Software Engineering for their encouragement and help.

A special thanks to the current and former graduate financial officers at Lakehead University graduate office Maegen Lavallee and Trish Sokoloski for their continuous support and assistance throughout my school years.

I would like to thank Lakehead University for its kind support in providing me with an excellent environment to conduct research.

Finally, I would like to thank NSERC and Qwantech for their collaborative financial support in my early years of research through the NSERC IPS2 grant.

Abstract

Various applications in computational linguistics and artificial intelligence rely on high-performing semantic similarity and word sense disambiguation techniques to solve challenging tasks such as information retrieval, machine translation, question answering, and document clustering. While text comprehension is intuitive for humans, machines face tremendous challenges in processing and interpreting a human's natural language. This thesis discusses two interconnected natural language processing tasks using a contextual semantic approach and knowledge-based repository. The first task is a knowledge-based semantic similarity and relatedness between words using WordNet, and the second is a knowledge-based semantic word sense disambiguation. The semantic similarity and relatedness task determines the level of likeness and connectedness between two words within a given context based on their semantic representation within a knowledge graph. The word sense disambiguation task determines the correct sense (meaning) of a word within sentence and document contexts.

The main focus of current research in this field relies solely on the taxonomic relation "ISA" to evaluate semantic similarity and relatedness between terms. Semantic similarity and relatedness have not been exploited to their full potential to solve integral natural language processing tasks, such as the word sense disambiguation task. Despite the wide range of knowledge-based word sense disambiguation approaches, the underlying similarity measure for most of them is the word overlap measure (i.e., Lesk similarity measure), which is, by definition, limited to the exact match of terms between the compared texts. This thesis explores the benefits of using all types of non-taxonomic relations in WordNet knowledge graph to enhance existing semantic similarity and relatedness measures. We propose a holistic poly-relational approach based on a new relational-based information content and non-taxonomic-based weighted paths to devise a comprehensive semantic similarity and relatedness measure. Furthermore, we propose a novel knowledge-based word sense disambiguation algorithm, namely Sequential Contextual Similarity Matrix Multiplication algorithm (SCSMM). The SCSMM algorithm combines semantic similarity, heuristic knowledge, and document context to respectively exploit the merits of local

context between consecutive terms, human knowledge about terms, and a document's main topic in disambiguating terms. Unlike other algorithms, the SCSMM algorithm guarantees the capture of the maximum sentence context while maintaining the terms' order within the sentence. Also, we identify the core factors that affect our proposed algorithm and most existing word sense disambiguation systems.

The results of the proposed algorithms demonstrate an improvement over the benchmark methods, including the state-of-the-art knowledge-based techniques. Our proposed semantic similarity and relatedness measure demonstrated improvement gain that ranged from 3.8%-23.8%, 1.3%-18.3%, 31.8%-117.2%, and 19.1%-111.1%, on all gold standard datasets MC, RG, WordSim, and Mturk, respectively. On the other hand, the proposed SCSMM algorithm outperformed all other algorithms when disambiguating nouns on the combined gold standard datasets, while demonstrating comparable results to current state-of-the-art word sense disambiguation systems when dealing with each dataset separately. Finally, the thesis discusses the impact of granularity level, ambiguity rate, sentence size, and part of speech distribution on the performance of the proposed algorithm.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Acronyms

AW All-Words. 11, 76–78, 92

BiLSTM Bidirectional Long Short-Term Memory. 31

CBC Clustering by Committee. 34

CSM Contextual Similarity Matrix. 78, 80, 85, 86, 97

DFS Depth First Search. 38

DocCF Document Carry Forward. 88, 97

IC Information Content. 5–7, 11, 15–18, 48, 49, 51–56, 58, 65–67, 74, 75, 96, 105, 106

IDF Inverse Document Frequency. 39

IGF Inverse Glass Frequency. 39

IMS "it makes sense". 30

IR Information Retrieval. 5, 6, 14, 15, 26, 27, 93, 105

ISA IS A. 4, 17, 20, 45, 61

KB knowledge base. 15–17

KG Knowledge Graph. 1, 2, 4, 5, 7, 9, 10, 13, 15–17, 20, 22, 24, 34, 39, 40, 42, 43, 45, 47, 49, 51–54, 60, 62, 65, 78, 90, 96, 104–107

KGE Knowledge Graph Embedding. 16, 73, 75

LCS Least Common Subsumer. 21, 35, 49, 67

LDA Latent Dirichlet Allocation. 40–42, 107

LOD Linked Open Data. 2, 10, 47

LS Lexical Sample. 76, 77

LSA Latent Semantic Analysis. 39

LSTM Long Short-Term Memory. 30, 32

MFS Most Frequent Sense. 9, 37, 40, 90, 97

ML Machine Learning. 28, 29

MSE Mean-Squared Error. 56–58

MT Machine Translation. 6, 26

NER Named Entity Recognition. 26

NLP Natural Language Processing. 1, 4–7, 9, 14, 15, 26, 28, 30, 78, 80, 95, 105

NLTK Natural Language Toolkit. 73, 96

NN Neural Network. 16, 30

OMSTI One Million Sense-Tagged Instances). 7, 11, 28, 37, 40, 81, 83, 90, 91, 96, 97

POS Part Of Speech. 32, 61, 62, 80, 90, 92, 96–104, 106

PR-SSR Poly-Relational Semantic Similarity and Relatedness. 47, 48, 53, 54, 66, 94, 95, 105, 106

QA Question Answering. 26, 27

RDF Resource Description Framework. 48, 65

RIC Relation Information Content. 49, 52, 55, 56, 58, 65, 68, 105

SCSMM Sequential Contextual Similarity Matrix Multiplication. 76, 78, 85–87, 90, 94, 95, 97, 99, 103, 104, 106, 107

SemIC Semantic Information Content. 48, 54–56, 67–69, 75

SG semantic graph. 2, 47

SSD Summation of Squared Difference. 56

SVM Support Vector Machine. 30

TF-IDF Term Frequency-Inverse Document Frequency. 27, 39, 78

WSD Word Sense Disambiguation. 1, 6, 7, 9–12, 14, 15, 21, 25–28, 30, 33, 35, 37, 38, 40, 44, 45, 76–78, 80, 85, 90, 91, 93–95, 99–101, 104, 106, 107

WSI Word Sense Induction. 32

Chapter 1

Introduction

While text comprehension is intuitive for humans, machines face tremendous challenges in processing and interpreting human natural language. The research area of Natural Language Processing (NLP) deals with the study of the computer's ability to process, analyze, and extract meanings from human's natural language. NLP has been the focus of the research community for many decades. However, until now, many NLP tasks have yet to be solved, such as sentence boundary detection, concepts similarity and relatedness, Word Sense Disambiguation (WSD), sentence similarity, topic detection, text summarization, and text generation. The main objective of any NLP task is to allow machines to achieve an automatic level of processing and handling of language as close as possible to a human's. This thesis discusses two interconnected NLP tasks using a contextual semantic approach by exploiting a knowledge-based repository. The first task is semantic similarity and relatedness between words, and the second is a knowledge-based semantic WSD task. The semantic similarity and relatedness task determines the level of likeness and connectedness between two words within a given context based on their semantic representation within a Knowledge Graph (KG). The WSD task determines the correct sense (meaning) of a word within sentence and document contexts.

In this thesis, we propose a comprehensive semantic similarity and relatedness measure that exploits the KG structure to its full potential. Furthermore, we propose a novel WSD approach that employs knowledge-based semantic similarity and relatedness mea-

sures. This chapter starts with an overview of the area of study, Section 1.1. Then, Section 1.2 describes the specific problems and research challenges addressed in this thesis. In Section 1.3, we summarize the main contribution of our research. Finally, Section 1.4 describes the organization of the thesis.

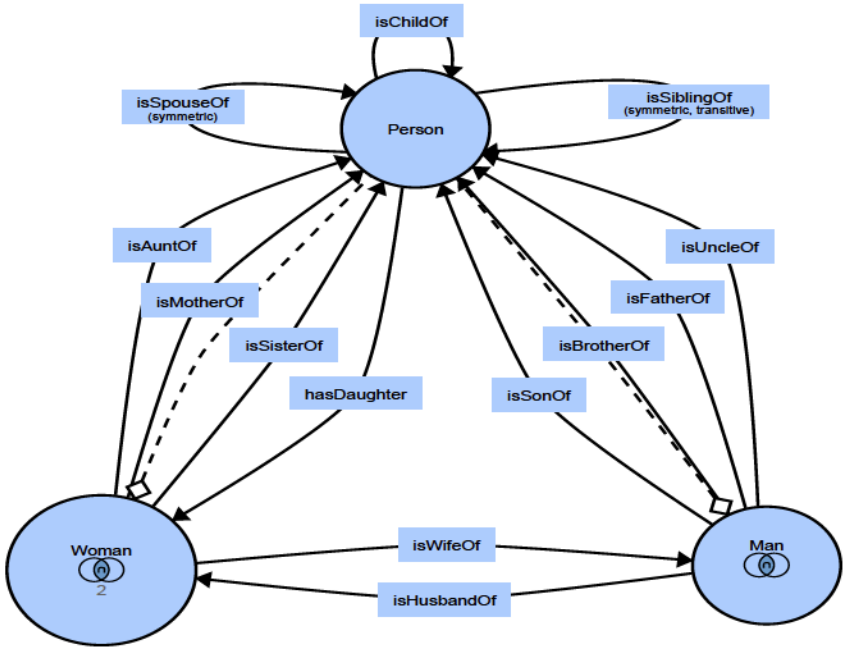
1.1 Overview

Data can be connected through various types of relations to present useful information. At the same time, a coherent collection of information produces specific knowledge. Linked Open Data (LOD) is a structure to allow linking data from multiple sources via meaningful semantic relationships to form useful information. Ontologies are a good example of LOD. Ontologies encompass a formal and machine-readable representation of entities' names, definitions, properties, and contextual relationships using formal links/relations that reflect a true connection in the real world [2]. Ontologies are sometimes referred to as KG. Examples of well known KGs include, but not limited to, DB bedia[1], WordNet[2], BabelNet[3], GeoNames[4], the Gene Ontology GO[5]. In this research, we use WordNet KG as our main lexical database.

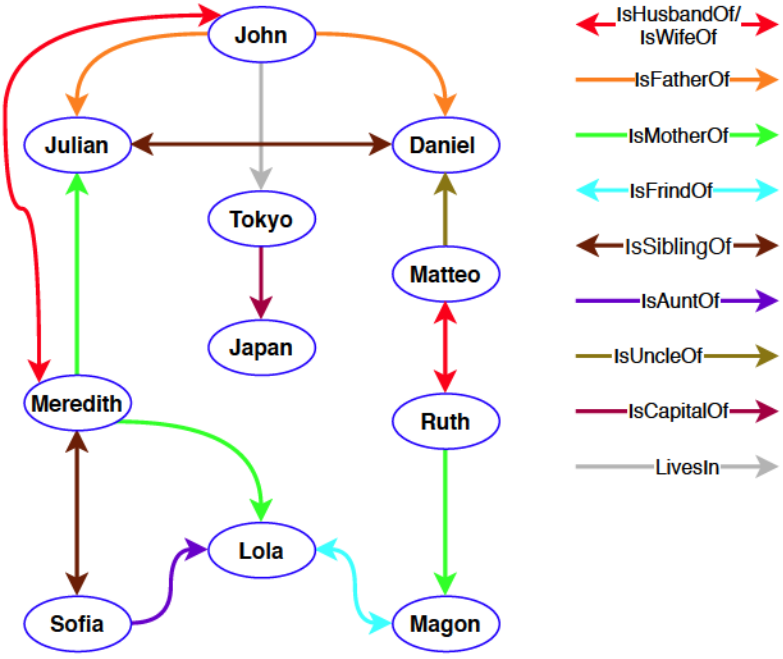
Gruber defined ontology as *"an explicit specification of conceptualization"* [2]. In technical terms, an ontology is a formal semantic representation of the concepts within a specific domain. The semantic representation is established through a set of axioms. An axiom connects two concepts and/or instances through a specific relation that models real-world connection in the form of *subject*, *predicate*, and *object*. For instance, *"John lives in Tokyo"* would be represented in the KG as *subject:John*, *predicate:LivesIn*, and *object:Tokyo*, see Fig. 1.1b. An interconnected set of axioms forms a KG or semantic graph (SG) as referred to in [3, 4, 5]. Ehrlinger and Wöß have formally defined KG as an acquisition and integration of information into ontology with a reasoner to derive new

[1] `https://wiki.dbpedia.org/`
[2] `https://wordnet.princeton.edu/`
[3] `https://babelnet.org/`
[4] `https://www.geonames.org/`
[5] `http://geneontology.org/`

knowledge [3]. An illustration of such a graph connecting instances and an ontology connecting concepts can be observed in Fig. 1.1b and Fig. 1.1a, respectively.



(a) Segment of ontology[6]



(b) Segment of KG

Figure 1.1: Segments of the family ontology and KG

[6]Visualization can be accessed at `http://vowl.visualdataweb.org/webvowl.html`

Since ontologies are domain-specific by nature, they incorporate a set of non-taxonomic relations that are relevant to the modeled domain. These relations resembles the semantic knowledge for all entities of that domain. Fig. 1.1 demonstrates such semantic representation, as it describes the family ontology through semantic relations that model real-world family relationships such as husband, child, spouse, and sibling.

Semantic is the study of meaning, a branch of the philosophical theory semiotics dealing with the relationships between signs and their meanings. Hebeler has described it as follows: *"Semantic gives a keyword symbol useful meaning through the establishment of relationships"* [6]. In computer science, it is the study of relationships between modeled entities within an ontology or KG. Whereas in NLP, semantic is the study of contextual relationships between words, sentences, or documents by analyzing asserted and inferred relationships within a KG. For instance, Fig. 1.2 depicts a sample of a KG presenting the taxonomic relations (i.e., IS A (ISA)) and some of the non-taxonomic relations between its concepts. *Eat* is an example of a non-taxonomic relation associating *Farm Animal* with *Vegetables*. ISA is the only mandatory relation, as it shapes the taxonomic structure of all entities within the KG. The non-taxonomic relations provide additional semantic information that represents real-world associations between entities within the modeled domain. The more non-taxonomic relations are present in the KG, the semantically richer the graph.

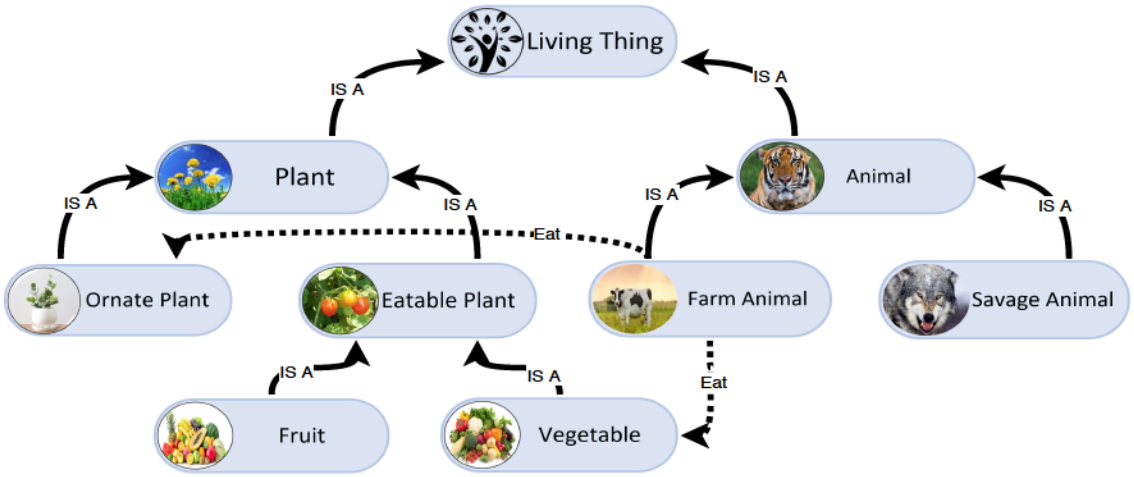


Figure 1.2: Semantic relations within a knowledge graph

Given the above definition, semantic relations have been a core element in the many research topics, especially within NLP domain. Semantic relations within KGs can determine the contextual similarity and relatedness between two given words or their concepts. Semantic relations can also be used to disambiguate words within a given context; this is a fundamental NLP challenge machines have yet to master. Finally, semantic knowledge can capture the contextualized similarity between two full sentences, which can be adopted to solve other Information Retrieval (IR) and NLP tasks such as topic detection, document clustering, classification, and document summarization.

In addition to the semantic relations, concepts within KG are considered informative entities, where each concept contains a specific amount of information that reflects its distribution within its domain. This notion is referred to as Information Content (IC). IC is a basic quantitative measure of the amount of information in something. It is derived from the probability of a particular event occurring from a random variable. Within a KG, concepts have been assigned an IC value using two approaches. The first relies on an external corpus and uses a probabilistic model. This approach is referred to as IC-extrinsic approach. The second depends on various features from within the KG, such as; child nodes, depth, leaves, and parents. This approach is referred to as IC-intrinsic approach. In this thesis, we focus on the IC-intrinsic approach.

1.1.1 Semantic Similarity and Relatedness

Semantic similarity and relatedness measure the level of likeness and connectedness between two terms based on their relations. Some literature evaluates the similarity and relatedness as a single distance measure between the meanings of two terms. On the other hand, others distinguish similarity as a specific case of relatedness, and relatedness is a more general measure. Hence, semantic similarity is a specific measure of likeliness, while relatedness is a more general measure that reflects connectedness. In this context, similarity can be measured by the taxonomic relations, while relatedness includes all other non-taxonomic relations. For example, in Fig. 1.2 the concepts *'Vegetable'* and *'Fruit'* will have high semantic similarity while *'Fruit'* and *'Savage Animal'* will have a

low semantic similarity. An example of high relatedness from the same ontology is the relationship between *'Vegetable'* and *'Farm Animal'* as they have a direct non-taxonomic relation (*'Eat'*) between them.

A wide range of semantic similarity measures have been proposed and applied in various applications and domains. These measures vary in performance based on their approaches and application domains. Detailed comparisons of these measures can be found in [7, 8, 9, 10, 11, 12, 13]. In summary, semantic similarity measures can be categorized into four main categories based on their approach: path, feature, IC, and hybrid. More details of these are presented in Chapter 2. However, the focus of this thesis is on IC-based semantic similarity and relatedness measures.

Semantic similarity and relatedness can be applied to solve challenging tasks, including the WSD that we present in the following section. Furthermore, semantic similarity can be applied in other NLP tasks, including text classification, information retrieval, machine translation, and document clustering. [9].

1.1.2 Word Sense Disambiguation

WSD is considered one of the oldest tasks of computational linguistics going back to the 1940s. It started as a distinct task since the beginning of machine translation. The first challenge that triggered WSD task is Machine Translation (MT) in the 1940s. Since then, researchers have been developing models and algorithms to improve the accuracy of this task using various approaches; supervised, semi-supervised, and knowledge-based systems. WSD is an essential task in many other applications, such as IR, information extraction, knowledge acquisition, and NLP. With the introduction of supervised machine learning in the 1990s, various supervised approaches attempted to solve the WSD task. More recent studies are exploring semi-supervised and unsupervised approaches by going back to using knowledge base in the form of graph systems such as WordNet[7], and BabelNet[8].

[7] `https://wordnet.princeton.edu/`

[8] `https://babelnet.org/`

Human beings can detect the appropriate sense unconsciously by large, whereas teaching that to a machine is challenging. Within the NLP domain, WSD is the task to determine the appropriate meaning (sense) of words given a surrounding context. WSD is considered a classification task, where the system's main task is to classify a specific word to one of its senses as defined by a lexical dictionary. One typical example is the word *'Bank'*, where it has eighteen different senses defined in WordNet[9] lexical database. Out of which, ten are defined as nouns, and the rest are defined as verbs, as shown in Fig 1.3.

Based on their approaches, WSD systems are divided into four main categories: supervised, semi-supervised, unsupervised, and knowledge-based. Supervised systems require a large sense-annotated training dataset, which is challenging to construct. To our knowledge, there are only two datasets available; the first is the SemCor dataset, which consists of 226,040 manually annotated senses divided into 352 documents [14]. The second is the One Million Sense-Tagged Instances) (OMSTI) dataset, which consists of one million automatically annotated senses [15]. The dataset was constructed based on a large English-Chinese corpus using an alignment-based WSD technique [16]. Various supervised systems have been designed to date. These systems use different techniques such as decision list [17], decision trees [18], naive bayes [19], and various neural network and sense embedding systems [20, 21, 22, 23, 24].

Semi-supervised systems employ a bootstrapping process with a small seed of a sense-annotated training dataset and a large corpus of un-annotated senses. At first, a supervised classifier is trained using the seed, then the iterative bootstrapping process gradually increases the size of the annotated dataset and shrinks the un-annotated one [25].

Other systems followed an unsupervised approach by using a context clustering [26], word clustering [27], and other graph-based algorithms such as *PageRank* algorithm [28].

The last category, which is the focus of our research is knowledge-based. Systems of this category rely on the structure and features of a KG. Different systems exploit various features, such as taxonomic relations, non-taxonomic relations, concept's IC, and paths.

[9] `http://wordnetweb.princeton.edu/perl/webwn?s=bank`

Word to search for: bank Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- S: (n) **bank** (sloping land (especially the slope beside a body of water)) *"they pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents"*
- S: (n) depository financial institution, **bank**, banking concern, banking company (a financial institution that accepts deposits and channels the money into lending activities) *"he cashed a check at the bank"; "that bank holds the mortgage on my home"*
- S: (n) **bank** (a long ridge or pile) *"a huge bank of earth"*
- S: (n) **bank** (an arrangement of similar objects in a row or in tiers) *"he operated a bank of switches"*
- S: (n) **bank** (a supply or stock held in reserve for future use (especially in emergencies))
- S: (n) **bank** (the funds held by a gambling house or the dealer in some gambling games) *"he tried to break the bank at Monte Carlo"*
- S: (n) **bank**, cant, camber (a slope in the turn of a road or track; the outside is higher than the inside in order to reduce the effects of centrifugal force)
- S: (n) savings bank, coin bank, money box, **bank** (a container (usually with a slot in the top) for keeping money at home) *"the coin bank was empty"*
- S: (n) **bank**, bank building (a building in which the business of banking transacted) *"the bank is on the corner of Nassau and Witherspoon"*
- S: (n) **bank** (a flight maneuver; aircraft tips laterally about its longitudinal axis (especially in turning)) *"the plane went into a steep bank"*

Verb

- S: (v) **bank** (tip laterally) *"the pilot had to bank the aircraft"*
- S: (v) **bank** (enclose with a bank) *"bank roads"*
- S: (v) **bank** (do business with a bank or keep an account at a bank) *"Where do you bank in this town?"*
- S: (v) **bank** (act as the banker in a game or in gambling)
- S: (v) **bank** (be in the banking business)
- S: (v) deposit, **bank** (put into a bank account) *"She deposits her paycheck every month"*
- S: (v) **bank** (cover with ashes so to control the rate of burning) *"bank a fire"*
- S: (v) count, bet, depend, swear, rely, **bank**, look, calculate, reckon (have faith or confidence in) *"you can count on me to help you any time"; "Look to your friends for support"; "You can bet on that!"; "Depend on your family in times of crisis"*

Figure 1.3: Senses for the term bank

The first knowledge-based system was developed by Lesk, based on the term's definition overlap with its sentence [29]. The Lesk model was extended to include the definitions of semantically related terms [30]. Other techniques of this category include the one we propose in this thesis: a semantic similarity technique. Each sense of the ambiguous word is assigned a weight based on its semantic similarity with other terms within the sentence, document, or both. The sense with the highest weight is selected as the correct sense.

The most straightforward approaches out of all are the two baselines approaches. The first is the Most Frequent Sense (MFS) baseline, which is a heuristic approach that selects the sense that appears the most within a training dataset. The second is the WordNet 1st sense approach, which is merely selecting the first sense as it appears in WordNet.

1.2 Problem Statement and Motivation

An overwhelming number of semantic similarity measures have been proposed in the literature. Some researchers considered similarity to be a specific case of relatedness [9, 31], while others did not distinguish between semantic similarity and relatedness [32, 33]. Nonetheless, for the majority of these methods, similarity has been evaluated strictly based on hierarchical relations (i.e., hyponym/hypernym), except for a few methods that have exploited a limited number of non-taxonomic relations to compute relatedness between concepts (i.e., meronymy/holonymy and antonymy) [9, 34, 35]. Even for those who adopted non-taxonomic relations, they treated all relations equally without analyzing their meanings and the information they carry between concepts.

Furthermore, to our knowledge, semantic similarity and relatedness measures have not been exploited to their full potential to solve integral NLP tasks, such as the WSD. Amongst all four WSD categories, supervised and knowledge-based are the most promising approaches [36]. However, supervised approaches require a large annotated dataset, which is challenging to produce. Due to the limited number of sense-annotated datasets, these systems face challenges to excel and demonstrate a noticeable improvement over other systems. Moreover, supervised systems need to be well trained, which is computationally and time expensive. Finally, most WSD supervised systems are unable to intuitively explain their results since they usually use a training function that leads to a calculated decision-making process.

On the other hand, knowledge-based systems do not require a training dataset, as they rely on a massive dictionary or KG. Moreover, knowledge-based systems can easily explain their results since they normally follow an intuitive process. With the advancement

of LOD and domain-specific KGs, these systems have a higher potential to outperform other approaches due to the advantage of broader KG coverage [37]. The two required ingredients to achieve that, are a semantically rich KG and a comprehensive semantic similarity and relatedness measure.

The limitations in current semantic similarity and relatedness measures and WSD systems motivated us to pursue the following objectives:

- Address the limitations in existing IC-intrinsic and path-based similarity methods.
- Study the effect of various non-taxonomic relations on enhancing semantic similarity and relatedness measures using WordNet KG.
- Design a new method that combines semantic similarity and relatedness into a single comprehensive measure that complements existing taxonomic measures.
- Investigate the effect of semantic similarity and relatedness measures, word sense heuristic, document context, and average sentence size on disambiguating words.
- Propose a new algorithm that exploits semantic similarity and relatedness, word sense heuristic, and document context to solve all-word WSD task.
- Evaluate our approaches using gold-standard benchmarks and state-of-the-art methods to demonstrate their robustness and scalability.

We tackled the above-mentioned objectives by adopting the methodology given below.

- Exploit all non-taxonomic relations within WordNet KG to design a new relational-based similarity and edge-weighted relatedness measures.
- Develop a framework to systematically evaluate and demonstrate the effect of the proposed relational-based similarity and relatedness on current benchmark methods.
- Develop a framework to show the effect of the following three parameters on WSD:

  - Semantic similarity and relatedness.
  - Word senses heuristics from SemCore and OMSTI datasets.
  - Document context.

- Formulate several experimental scenarios to validate all demonstrated results.

1.3 Contributions

Compared to the aforementioned literature and motivated by the importance of the researched topic, the contributions of this work can be summarized as follows:

- We propose a new relation-weighting schema based on the IC difference between linked concepts to measure non-taxonomic relational-based similarity.
- We propose a new relatedness measure based on a new non-taxonomic edge-weighted paths between terms.
- We propose a holistic poly-relational approach that exploits all non-taxonomic relations, their types and their frequency to enhance semantic similarity and relatedness in the context of WordNet.
- We propose a novel knowledge-based WSD technique that reflects human thinking by exploiting semantic similarity and relatedness, word sense heuristic, and document context for solving All-Words (AW) WSD task.
- We demonstrate the effect of various semantic similarity and relatedness measures, word sense heuristic, and document context on the performance of WSD methods.

1.4 Thesis Organization

Chapter 2 provides a preliminary study of the semantic similarity and relatedness measures. The chapter also presents background information about the WSD task. This chap-

ter also presents a comprehensive literature review of the related approaches and techniques. In Chapter 3, we present our proposed comprehensive semantic similarity and relatedness measure and its main parameters. Chapter 4 presents a novel knowledge-based WSD algorithm that employs semantic similarity, word sense heuristic, and document context. Finally, the conclusion of this thesis and future work are presented in Chapter 5

1.5 List of Publications

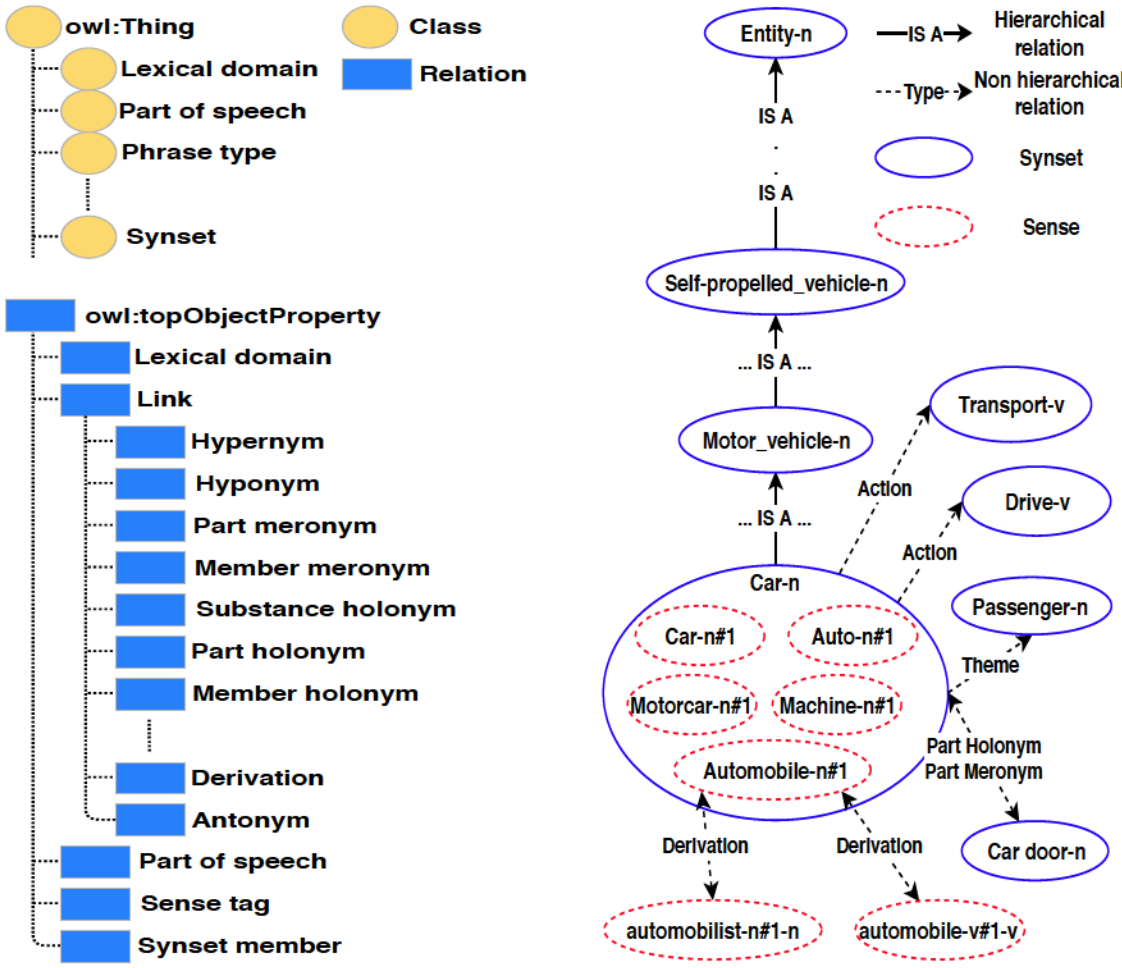
- M. AlMousa, R. Benlamri and R. Khoury, "Exploiting Non-Taxonomic Relations for Measuring Semantic Similarity and Relatedness in WordNet", Journal of Knowledge-based Systems, Elsevier, Nov 2020.
- M. AlMousa, R. Benlamri and R. Khoury, "A Novel Words Sense Disambiguation Approach using WordNet Knowledge Graph," Journal of Computer Speech and Language, Submitted, 2020.
- M. AlMousa, R. Benlamri and R. Khoury, "NLP-Enriched Automatic Video Segmentation," 6th International Conference on Multimedia Computing and Systems (ICMCS), Rabat, 2018, pp. 1-6, doi: 10.1109/ICMCS.2018.8525880.

Chapter 2

Background and Literature Review

2.1 Knowledge Representation

Ontologies and KGs are knowledge repositories that systematically model real world entities and their relationships in machine readable format. The semantic representation consists of a set of axioms that is a subject, predicate, and object. The subject and object are concepts with a relationship referred to as predicate. A complete set of axioms form an ontology or KG. Ontologies and KG are generally domain specific. Therefore, they incorporate a wide range of taxonomic and non-taxonomic relationships that models a specific domain context between concepts. The research community have been using the terms interchangeably, However, for the purpose of this research, we follow the mainstream understanding of KG as a representation of instances of ontological concepts for a specific domain [3]. To illustrate this definition, Fig. 2.1a describes the WordNet ontology, which includes concepts and object properties. The latter are referred to as links, pointers, or relations. On the other hand, Fig. 2.1b shows instances of concepts and their relations based on the designed ontology and inferred knowledge. Despite various definitions of KG and ontology, the terms have been used interchangeably when referring to some of the well known knowledge base repositories such as DBpedia [38], Freebase [39], YAGO [40], BabelNet [41], and WordNet [42] and similarly structured databases [9, 32, 34, 43].



(a) Sample ontology (b) Sample graph

Figure 2.1: A fragment of WordNet ontology and graph

In this thesis, we focus on WordNet as our main knowledge repository. WordNet is an English words lexicon database, that organizes concepts into a conceptual hierarchy. It was designed to semantically model English words through the categorization of synonyms and existing taxonomic and non-taxonomic relations [42]. Since the creation of WordNet, it has become a valuable resource used in many domains, including NLP, IR, and semantic-based recommender systems. Its semantic structure triggered the research community to examine tasks such as the ones we investigate in this research, namely semantic similarity and relatedness and semantic WSD. The next two sections present the related work for the studied topics.

2.2 Semantic Similarity and Relatedness

Semantic similarity is a measure of likeness between various text components such as words, sentences, or documents. It has a significant role in many NLP tasks, such as IR [44], text clustering [45], text classification [46], text summarization [47], and WSD [48].

Semantic similarity is also applied in recommender systems [49], geo-informatics [50], and biomedical informatics [51] domains. Semantic relatedness is a measure of the contextual relationship between words, sentences, or documents. It is a more general measure than semantic similarity, as two dissimilar words can be very related; for example, 'bird' and 'feather' are conceptually dissimilar yet are intuitively related. Nonetheless, most literature has used the two measures interchangeably. Semantic similarity and relatedness measures can be categorized into two main categories based on their knowledge resources. The first is corpus-based measures, which include statistical approaches [52], neural network [53], and word embedding approaches [54, 55, 56, 57]. The second category is knowledge base (KB) measures, which uses an ontology or KG structure to measure the similarity and relatedness between terms. Researchers have also categorized similarity and relatedness measures based on other criteria, such as their technique (supervised or unsupervised), topological, and statistical. A more detailed comparisons of such categorization can be found in [7, 8, 9, 10, 11, 12, 13].

2.2.1 Semantic Similarity Methods

This thesis focuses on the KB approaches, which can be further categorized into four main categories: path, feature, IC, and hybrid measures.

Path-based: These measures count the number of edges in the shortest path between concepts. The longer the path between two concepts, the less similar they are, and vice versa [33, 35, 58, 59, 60, 61].

Information Content-based: Based on the information source, IC-based measures can be either extrinsic or intrinsic.

- **Extrinsic**: Extrinsic IC measures are corpus-based, meaning an external corpus and a statistical model are used to compute the information of each concept [52, 62, 63].
- **Intrinsic**: Intrinsic IC measures, are KG-structure-based, meaning the concept's information lies within the KG topological structure. Various structural attributes have been used as indicators of the information contained within each concept [9, 32, 33, 34, 43, 64, 65].

Feature-based: Feature-based measures represent a concept as a vector of features constructed from its attributes [66, 67, 68, 69]. Jiang et al. formally represented Wikipedia concepts as a structured knowledge base and proposed a multi-vector feature-based approach that includes features from concept's synonyms, glosses, Anchors, and Categories [70]. Recent methods in this category incorporated Neural Network (NN) models to embed feature vectors that represent the entities, relations, and the entire KG. These methods are being referred to as Knowledge Graph Embedding (KGE) [57, 71].

Hybrid: Finally, hybrid similarity measures combine two or more of the above [8, 33, 61, 69, 72].

From the above-mentioned approaches, we are more interested in the semantic similarity and relatedness measures that are IC-based. In particular, we focus on the IC-intrinsic approaches. The IC-intrinsic approaches compute the concepts' IC value based on various graph-based features such as the number of hyponyms, depth, siblings, leaves, and other graph features. On the other hand, IC-Extrinsic approaches employ a statistical approach on an external corpus to compute the information contained within each concept that exists in the KG.

It is worth pointing out that some KB approaches have viewed similarity and relatedness as a single measure based on various topological features. On the other hand, some

Table 2.1: IC-intrinsic measures

IC Measures	Formulae	Hierarchical features
Seco [32]	$ic_{seco}(c) = 1 - \frac{log(hypo(c)+1)}{log(max_{wn})}$	hyponyms
Zhou [43]	$ic_{zhou}(c) = k\left(1 - \frac{log(hypo(c)+1)}{log(max_{wn})}\right) + (1-k)\left(\frac{log(deep(c))}{log(max_{deep})}\right)$	hyponyms, depth
Sebti[1][33]	$ic_{sebti}(c) = -log \prod_{c_i \in hyper(c)} \frac{1}{DirHypo(c_i)}$	hypernyms, direct hyponyms
Meng [65]	$ic_{meng}(c) = \frac{log(deep(c))}{log(max_{deep})} \times \left(1 - \frac{log\left(\sum_{a \in hypo(c)} \frac{1}{deep(a)} + 1\right)}{log(max_{wn})}\right)$	depth, hyponyms' depth
Sánchez [34]	$ic_{s'anchez}(c) = -log\left(\frac{\frac{\lvert leaves(c)\rvert}{\lvert subsumers(c)\rvert}+1}{max\_leaves_{wn}+1}\right)$	leaves, hypernyms
Cai [9]	$ic_{cai}(c) = \left(1 - \frac{log(hypo(c)+1)}{log(max_{wn})}\right) \times tanh(deep(c))$	hyponyms, depth
Zhang [64]	$ic_{zhang}(c) = K\left(1 - \frac{log(hypo(c)+1)}{log(max_{wn})}\right) - (1-K)\frac{1}{n}\sum_{i=1}^{n} log\,(\omega)$	hyponyms, hypernyms' siblings

where $\omega = \prod_{c_i \in hyper(c)} \frac{1}{sibling(c_i)} + 1$,
$K = \frac{hypo(c)}{hyper(c)+hypo(c)}$,
n is number of direct parents

other KB approaches viewed similarity as the conceptual likeness between terms based on a single taxonomic relation (ISA), whereas relatedness is based on all other relations. This research adopts a single similarity and relatedness measure based on all existing relationships within the KG. Also, it extends the concept of IC to all relationships and concepts within the KG.

The next two subsections present in detail the literature related to IC-intrinsic based approaches and approaches that exploited non-taxonomic relationships.

2.2.2 IC-Intrinsic Approaches

In this section, we further discuss the IC-intrinsic-based semantic similarity and relatedness measures, which had heretofore been presented by the research community. These are used as benchmarks to evaluate the proposed method in this thesis. Table 2.1 lists the IC-intrinsic measures implemented through various topological features of WordNet.

Seco [32] was the first to introduce an intrinsic IC measure that is not dependent on an external corpus. His approach relies on the intrinsic features of the KG, specifically the number of hyponyms within the concept. He proposed an IC-intrinsic measure as a monotonically decreasing function with the number of hyponyms for a given concept. Seco's IC-intrinsic model proved that the number of hyponyms inversely conveys the concept's IC [32].

Using Seco's IC, Zhou [43] and Cai [9] incorporated the concept's depth to emphasize the generalization/specialization effect on IC. Both approaches used depth to overcome Seco's method's limitation of attributing concepts with equal IC values regardless of their hierarchical level in the taxonomy. Zhou introduced a new IC measure as a function of normalized depth and hyponyms to compute the concept's IC [43], Cai proposed a new IC measure as a nonlinear transformation function to measure the contribution of depth to the concept's IC. Furthermore, Cai proposed a similarity measure to evaluate the IC measure [9].

Sebti proposed a new IC-intrinsic measure as a monotonically-increasing function of depth and number of siblings. He utilized the branching factor of all subsumers as an indicator of information gained through ancestor concepts. Hence, his new measure incorporated the number of subsumers with the probability of branching using direct hyponyms. Sebti's measure is not normalized, thus, the IC values could have the range of $[0, \infty)$. Furthermore, he improved his IC measure with an edge-counting tuning semantic similarity function [33]. This approach clearly confirms the parent-child effect on IC, fol-

[1]The authors did not explicitly state the final equation in their article. However, they demonstrated it through an example as follow: $IC(Box) = -Log\left(\frac{1}{9} \times \frac{1}{10} \times \frac{1}{36} \times \frac{1}{42} \times \frac{1}{13} \times \frac{1}{49}\right) = 18.2778$, where the denominator represents the number of siblings from the highest subsumer to the concept

lowing the inheritance principle, while being a monotonically decreasing function moving from leaf to root.

Inspired by information theory, Sánchez proposed a new leaf-based intrinsic IC measure. He argued that the concept's IC is directly proportional to its subsumers and inversely proportional to its leaves. Hence, his IC measure is described as a measure of concept's concreteness level to its abstraction level, specificity to generality. Unlike previous studies, Sánchez incorporated multiple inheritance in the semantic similarity measure through the number of subsumers [34].

In another study, Meng exploited the depth of a concept, as well as that of its hyponyms, in order to overcome Seco's approach of attributing the same IC value to all leaves. In other words, Meng's main argument was that leaves at a higher level of the taxonomy (i.e., smaller depth) convey less information than deeper leaves; hence, they have a smaller IC value [65].

Zhang introduced a new IC-intrinsic measure exploiting multiple inheritance. Zhang's improvement came from covering multiple inheritance concepts as well as incorporating the concept's siblings with depth, hyponyms, and hypernyms [64]. Another Multiple inheritance approach was recently developed by Hussain [73]. His approach utilizes a new neighbourhood ancestor semantic space to define concept's IC value. This technique is applied on a semi-structured taxonomy KG called Wikipedia Concept Graph (WCG) [73].

Table 2.1 lists the IC-intrinsic measures described above, which are implemented through various topological features of WordNet. These IC measures were evaluated using either existing similarity measure such as Resnik [52], Lin [63], and JCN [62], or new proposed similarity measure. For example, Cai [9] and Zhang [64] proposed new similarity measures exploiting the benefits of their new IC. Table 2.2 lists all pre-existing and new similarity measures based on IC-intrinsic measures.

Table 2.2: IC-based similarity measures

Sim Measures	Formulae
Resnik [52]	$sim_{res}(c_1, c_2) = \max_{c \in S(c_1,c_2)} ic(c)$
Lin [63]	$sim_{lin}(c_1, c_2) = \frac{2 \times sim_{res}(c_1, c_2)}{(ic(c_1) + ic(c_2))}$
JCN [62]	$sim_{jc}(c_1, c_2) = 1 - \frac{ic(c_1) + ic(c_2) - 2 \times sim_{res}(c_1, c_2)}{2}$
Cai [9]	$sim_{Cai_1}(c_1, c_2) = \exp^{-(\alpha \times spl_W(c_1,c_2) + \beta \times spl_N(c_1,c_2))}$
	$sim_{Cai_2}(c_1, c_2) = \exp^{-(\alpha \times spl_W(c_1,c_2) + \beta \times spl_O(c_1,c_2))}$
	$spl_W(c_1, c_2) = ic(c_1) + ic(c_2) - 2 \times ic(LCS(c_1, c_2))$
	$spl_N(c_1, c_2) = \frac{len(c_1,c_2)}{2 \times Max_{deep}}$
	$spl_O(c_1, c_2) = log\left(\frac{deep(c_1)+deep(c_2)+1}{2 \times dep(LCS(c_1,c_2))+1}\right)$
Zhang [64]	$sim_{zhang}(c_1, c_2) = 1 - log\left(2 - \frac{2 \times ic(LCS(c_1,c_2))}{ic(c_1)+ic(c_2)}\right)$

2.2.3 Non-Taxonomic Approaches

The majority of semantic similarity methods have been strictly evaluated based on the taxonomic relationship ISA (i.e., hyponym/hypernym within WordNet). However, few methods have exploited a limited number of non-taxonomic relationships (i.e., meronymy/holonymy and antonymy) to compute various relatedness measures [9, 34, 35].

WordNet includes many categories of non-taxonomic relations to enhance semantic similarity and relatedness measures that have yet to be exploited by researchers. Table 2.3 depicts theses relations and their usage frequency within WordNet KG. For instance, those methods which used meronymy relation did not distinguish between the three types of meronym relations: part of, substance of, and member of. This distinction is important because each meronym relation contributes different type of information to the semantic definition, and conveys specific information about the nature of the association between concepts. Based on the relation type (part of, substance of, and member of), the subject of an axiom conveys its inclusion in a larger entity, its physical components, or its participation in a group, respectively. Furthermore, other non-taxonomic relations that have

not yet been exploited, such as synonym, derivation, antonym, theme, cause, and action, convey an important informative component of a concept's semantic definition and information content. Therefore, a comprehensive semantic similarity measure should fairly incorporate information from all relations. Furthermore, relatedness in the literature was mostly evaluated based on the length of the path between two terms, without considering the importance of a particular relation within the modeled domain. We strongly believe that the most frequently used relations in a modeled domain, convey contextually related concepts in that domain.

Nonetheless, few studies used non-taxonomic relations to solve challenges such as spelling error correction and WSD based on the relatedness between concepts. In [30, 74, 75], the authors attempted to solve WSD by using a gloss vector. They evaluated a relatedness measure between concepts using standard vector-based similarity measures (i.e., overlap, cosine similarity). The vector's dimensions are words extracted from a concept's glossary in WordNet. However, the employed method was more of a linguistic/NLP approach rather than a semantic one, as they evaluated the English definition from a glossary rather than semantic relations.

In Liu [31], concepts are expressed by their relevant concepts as a vector. Relevancy is defined by the set of hypernyms and hyponyms. Dimensions are represented as their local densities (i.e., number of siblings in this case). To improve similarity and relatedness, the authors computed the relatedness strength between two concepts based on the number of paths between them. A path could be direct PartOf path (i.e., one concept is part of the other), or indirect (i.e., one concept is part of an element from the relevant set of the other). The relatedness strength is then added to the Least Common Subsumer (LCS) as a new sibling, and a taxonomic-based approach is applied to compute similarity [31]. A major limitation in this approach is that the paths are not pure relational, but mostly hierarchical. It nonetheless demonstrated that multiple paths are directly proportional to the relatedness strength, and can be employed to improve relatedness.

In [8], the authors explored a new path-based approach. $Path_{ISA}$ and $path_{PartOf}$ are computed based on ISA and PartOf relations, respectively. Then the shortest of the two

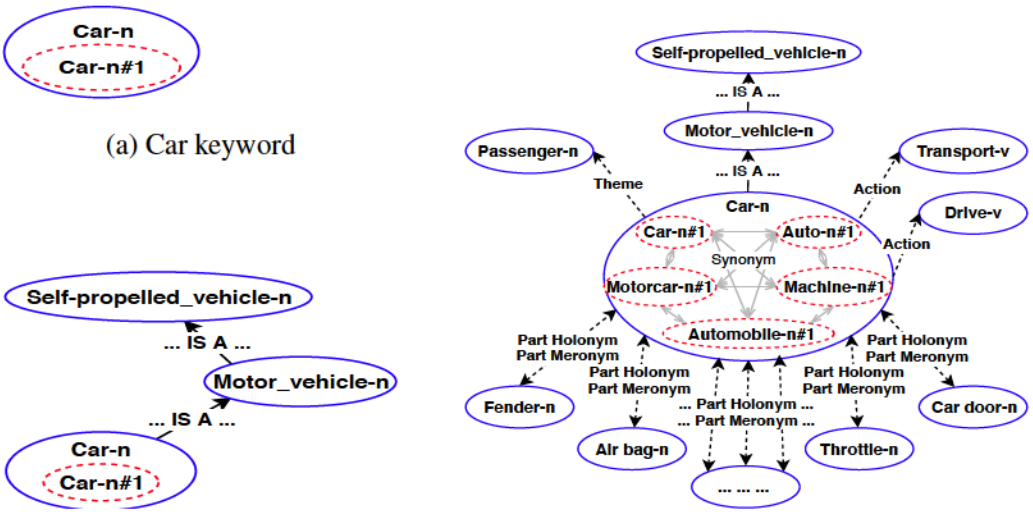
is selected to compute the similarity level between the two concepts. The main innovation of this method lies within the $path_{ISA}$ taxonomic approach. This is because the $path_{PartOf}$ is limited to a direct PartOf relation, or at most two such relations that connect two concepts through a common meronym [8]. This is a major limitation of this approach, especially, with a limited to none such paths exist between pairs within the used datasets.

Table 2.3: Non-taxonomic semantic relations in WordNet

Relation Name	Frequency	Prevalence
synset_member (synonym)	145076	74.61%
member_meronym	12252	6.30%
member_holonym	12242	6.30%
part_meronym	9082	4.67%
part_holonym	9071	4.67%
derivation	2957	1.52%
antonym	2154	1.11%
substance_holonym	746	0.38%
substance_meronym	744	0.38%
theme	103	0.05%
cause	15	0.01%
action	3	0.00%

2.2.4 Critical Analysis of the Related Work

Various intrinsic IC measures were proposed and used to determine the semantic similarity between concepts. As described in Section 2.2.2, these IC measures exploited different taxonomic features of the KG. A common limitation of these measures is that they rely solely on a single semantic dimension — the taxonomic 'is a' (ISA) relation — and that they ignore all other semantic dimensions, hence, limiting semantic similarity strictly to the generalization/specialization relation. However, by definition, *"Semantics give a keyword symbol useful meaning through the establishment of relationships"* [6]. This is clearly illustrated in the example shown in Fig. 2.2. The isolated first sense of the noun *Car*, denoted as *Car-n#1*, has no semantic meaning except that it is a member



(a) Car keyword

(b) Car with taxonomic relations

(c) Car with all semantic relations

Figure 2.2: The car concept in WordNet graph

of synset *Car-n* (see Fig. 2.2a). When adding the taxonomic semantic relation ISA, as shown in Fig. 2.2b, one can then elaborate on the definition as follows: *"Car is a motor vehicle, which is a self-propelled vehicle, etc."*. However, when we further include explicit (e.g. *part holonym /meronym, theme, action, etc.*) and implicit (i.e. synonym) non-taxonomic relations, the semantic definition of *Car-n#1* is significantly enriched, and hence its IC. The new definition of *Car-n#1* would be: *"Car is a motor vehicle, which is a self-propelled vehicle. Car has a theme as passenger. Car leads to actions such as transport and drive. Car has some parts such as car door, throttle, air bag, fender, etc. Car has synonyms (auto, machine, motorcar, and automobile)"*. To overcome this limitation, we argue that semantic relations other than ISA enhance the concept's semantic definition and increase its IC value; an increase that is proportional to the type and strength of that relation. Hence, this work extends the taxonomic-based IC principle applied in previous literature by exploiting non-taxonomic semantic relations.

As described in Section 2.2.3, it can also be observed that the few non-taxonomic approaches available in the literature do not exploit non-taxonomic relations to their full potential. They either limit the non-taxonomic relational path to one or two links or only rely on a single relation (i.e., PartOf).

Domain ontologies and KGs are contextually designed for specific purposes; there-

fore, they include a rich set of non-taxonomic relations that contextually describe the relationships between concepts. This is clearly illustrated in Fig. 1.1a, which describes the family ontology in terms of relations that model real-world family relationships such as husband, child, spouse, and sibling. On the other hand, Fig. 1.1b describes an example KG of family domain with many non-taxonomic relations describing various relationships between persons of the modeled family. Relatedness between persons of a family can be better evaluated using these relations, focusing on their types and occurrences in a path between two persons. Therefore, to overcome the limitations of existing non-taxonomic approaches, in this thesis, we consider the type and frequency of all non-taxonomic relations to measure relatedness between concepts. Thus, incorporating all contextually related concepts and privileging the dominant relations in the described domain. Consequently, in this thesis, we propose a weighted relational-based path approach, exploiting all non-taxonomic relations between two concepts to measure their relatedness and enhance their semantic similarity.

To summarize, various principles and findings from the aforementioned analysis of the literature motivated this research. Firstly, the role of relationships in ontology and KG design, and their importance to serve a contextual purpose within the modeled domain. Ontologies are domain-specific with a precise set of relations that enhances the semantic representation of the data within the KGs [6]. Therefore, ignoring these relations leads to an incomplete semantic evaluation. Secondly, our approach takes into consideration the information inheritance principle, such as the one used in the taxonomic approaches based on ISA relation. A child concept accumulates information from its parent concept(s), and adds its own new information. However, we strongly believe that other relations also contain and convey important information between concepts. Therefore, the effect of non-taxonomic relations on similarity is unavoidable. Finally, it should be noted that the benchmark datasets are measured by humans based on the combination of similarity and relatedness. Therefore, to provide fair evaluation and comparison with the gold standard, our approach makes use of all types of relations in a KG to devise a single measure that incorporates both the semantic similarity and relatedness between concepts.

[1] Visualization is done using `http://vowl.visualdataweb.org/webvowl.html`

Inspired by the related work, and motivated by the aforementioned drivers, we present the following observations:

Observation 1 *Considering non-taxonomic relations in a KG enhances information content, thus yielding more semantic similarity between concepts than just relying on taxonomic relations.*

Observation 2 *The prevalence of a non-taxonomic semantic relation within a KG is an indicator of its importance and relevance to the modeled domain. Thus, it has an impact on the relatedness between concepts.*

2.3 Word Sense Disambiguation

The main objective of WSD is to classify a word within a given context into its correct sense. This task has been investigated within the computational linguistics field since the 1940s, and since then, many algorithms and techniques have been developed. WSD is a challenging task for several reasons, one of which is related to the discrepancies of senses choices between dictionaries. One dictionary might provide more senses for a word than another. To overcome such a challenge, many researchers relied on a single comprehensive machine-readable lexical dictionary such as WordNet[2], Wikipedia[3], and BabelNet[4].

Another difficulty is derived from the evaluated test datasets and the inter-annotator agreement. The datasets to evaluate any system must be judged and annotated by humans because human judgment is considered a gold standard for any system. Compiling test datasets is not an easy task, as it is difficult for humans to remember or know all senses for all words, including their precise meanings and differences from other senses. The gold standard datasets usually measured by the inter-annotator agreement. Based on [37, 76, 77, 78, 79] the inter-annotator agreement using WordNet ranges between 67% and

[2] https://wordnet.princeton.edu/
[3] https://www.wikipedia.org/
[4] https://babelnet.org/

80% on fine-grained inventory. Such a low range of inter-annotator agreement encouraged the research community to develop and further investigate coarse-grained databases. In fact, some of the coarse-grained inventory has achieved up to 90% inter-annotator agreement [37, 79, 80]. Nonetheless, significant efforts has been made to compile high-quality datasets that are considered the primary gold standard for WSD systems (i.e., SensEval2, SensEval3, SemEval 2007, SemEval 2013, and SemEval 2015). These datasets are further discussed in Section 4.4.1.2.

2.3.1 Applications

Many NLP applications rely on WSD, either directly or indirectly. The list includes, but is not limited to MT, IR, Question Answering (QA), Named Entity Recognition (NER), text summarization, etc. Below we describe some of the most common applications, and Fig. 2.3 depicts more of such applications.

- **Machine Translation** [81, 82]: One of the main applications of WSD is MT, as it is required to determine the lexical choice in order to provide the appropriate translation. For instance, in a financial context, the English word '*change*' could be translated to French to either '*changement*' ('transformation') or '*monnaie*' ('pocket money').

- **Information Retrieval** [83, 84]: Accurate disambiguation of a search query can help prune some documents containing the same word but have a different context. Query expansion is one technique that employs WSD through a relevance feedback approach to improve IR performance. As the name suggests, the search query is modified by adding, removing, or reweighting the query words. The expansion can be achieved by an explicit or implicit WSD. An explicit WSD includes synonym words from the same Synset in WordNet. While implicit WSD includes the most frequent words that appear in a corpus. Paskalis and Khodra demonstrated the effect of successful WSD on the performance of an IR system. They concluded that a simple WSD approach, along with relevance feedback, improves the performance of IR

Figure 2.3: WSD in NLP

system [83]. In another research, Stokoe demonstrated an improvement of sense-based IR system over a traditional vector-based Term Frequency-Inverse Document Frequency (TF-IDF) approach.

- **Question Answering** [85]: WSD plays a critical role in QA systems. Imagine a system answering the following question '*When did George Bush enter the White House*?', this question is ambiguous as it is not clear which George Bush is being referred to here. Therefore, additional context might help disambiguate the question.

- **Named Entity Recognition** [86]: Some entities have the term 'bank' as part of its name, for instance, consider the following sentences:

  1. *I went to the **Bank of Montreal**.*

2. *I went to the **bank** of the Amazon river.*

3. *I went to the **bank** to cash a cheque.*

The first sentence has the word *bank* as part of the name of the financial institution **Bank of Monteral**. In contrast, the second sentence refers to land beside the Amazon river (the 1st noun sense in Fig. 1.3). Finally, the third sentence describes the second sense (the 2nd noun sense in Fig. 1.3) as the financial institute.

2.3.2 WSD Approaches

A vast number of research approaches, techniques and models have attempted to solve the WSD challenge as a standalone task or as part of a larger NLP application [37, 87, 88, 89, 90, 91, 92]. Either way, these approaches are grouped into four conventional categories:

2.3.2.1 Supervised Approach

Supervised techniques require the use of a training dataset (i.e., a sense-annotated corpus). However, these corpora are hard to produce due to the complexity of identifying the best combination of words' senses based on their definitions from WordNet. To our knowledge, there are currently two such datasets available: SemCor [14] and OMSTI [15], which will be discussed in Section 4.4.1.1. Supervised approaches also require a Machine Learning (ML) technique that will, through training, create a feature vector for each ambiguous word, train a classifier to appropriately assign the correct sense class to an ambiguous target word, and finally, test it using a dataset to evaluate the model [21, 93].

Early development of supervised WSD approaches include rule-based, probabilistic, or statistical models. Many comprehensive surveys have covered the mathematical details of each model in [37, 88, 89, 90, 91, 92]. Nonetheless, to list a few, the following are some of the common supervised WSD techniques:

Decision List : This is one of the first rule-based techniques that can be viewed as an ordered list of weighted 'if-else' rules. The rules are constructed based on a sense-annotated training dataset, where each word will have a list of senses associated with specific features that dictate its score and hence rule order. Finally, based on the word occurrence and its feature vector, the model checks the decision list and selects the sense with the highest feature score. Table 2.4 describes a list of rules in the form of (feature, class, score) for the word '**bank**'. However, despite their advantages, decision lists are known for over-fitting drawbacks; hence, they are outperformed by many recent ML techniques [17, 92, 94, 95].

Table 2.4: A Decision list example for the word *Bank*

Feature	Sense Class	Score
account with **bank**	Bank-2 (financial)	4.83
stand/V on/P ... **bank**	Bank-2 (financial)	3.35
bank of blood	Bank-5 (supply)	2.48
work/v ... **bank**	Bank-2 (financial)	2.33
the left/J **bank**	Bank-1 (river)	1.12

Decision Trees : Instead of a list, rules are presented in a binary tree that leads to the appropriate class decision leaf based on a "yes-no" answer to each rule. Quinkan was the first to introduce this model in [96] and then extended it in [97]. Fig. 2.4 depicts the decision tree rules for the word **bank**. Although they are simple to understand and can be presented in a human-readable format, Mooney concluded that they have been outperformed by recent ML models, due to their data sparsity, and unreliable predictions caused by the small training dataset [18, 37, 88].

Naive Bayes : As the name suggests, this probabilistic classifier is based on the Bayes' theorem. For an ambiguous word, the sense with maximum conditional probability given the contextual features is selected [19, 98].

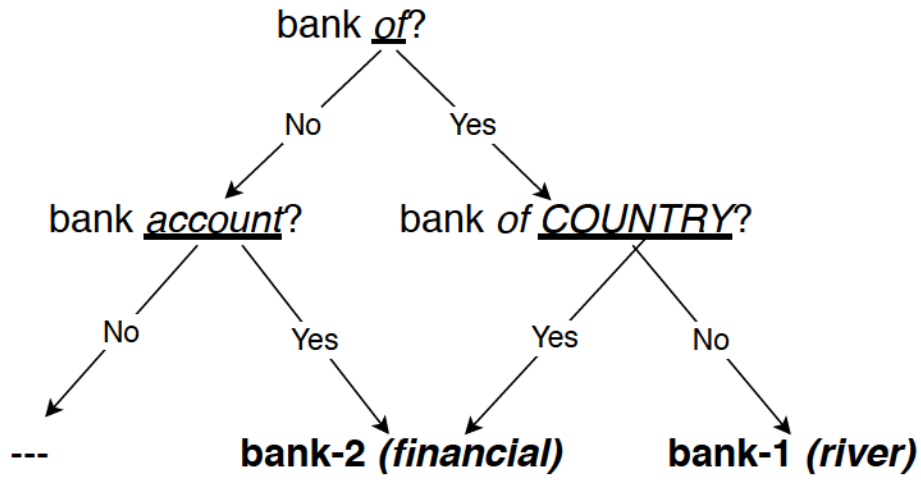


Figure 2.4: Decision tree example for the word *Bank*

Support Vector Machine : It was first introduced by Bose et al. in 1992. The goal of the Support Vector Machine (SVM) algorithm is to maximize the distance between positive and negative examples by learning a linear hyperplane from the training dataset [99]. Although SVM is a binary classifier, however, it can be used to separate each sense as one class from all other senses, and the sense with the largest class will be selected as the correct sense, Fig. 2.5 shows the selection of the best hyperplane between two classes based on two features. Although many researchers employed the SVM model to solve WSD [100, 101, 102], yet the "it makes sense" (IMS)[5] system is considered one of the first comprehensive publicly available systems that uses SVM for WSD [103].

Neural Network and Sense/Concept Embedding : NN has been employed in the field of NLP and, in particular, WSD since the early days. However, since the introduction of Word2vec in 2013 [54], the majority of research moved towards word, concept, and sense embedding, such as in [20, 21, 22]. One of the first incorporations of Mikolov et al. [54] and the original IMS [103] was the IMS+Word2vec system introduced by Iacobacci et al. [104]. Then, Papandrea proposed an improvement over both the original IMS system and the previous IMS+Word2vec [105]. The work in [23, 24] adopted a context2vec embedding on the original Long Short-Term Memory (LSTM) in [106] and

[5] `https://www.comp.nus.edu.sg/~nlp/software.html`

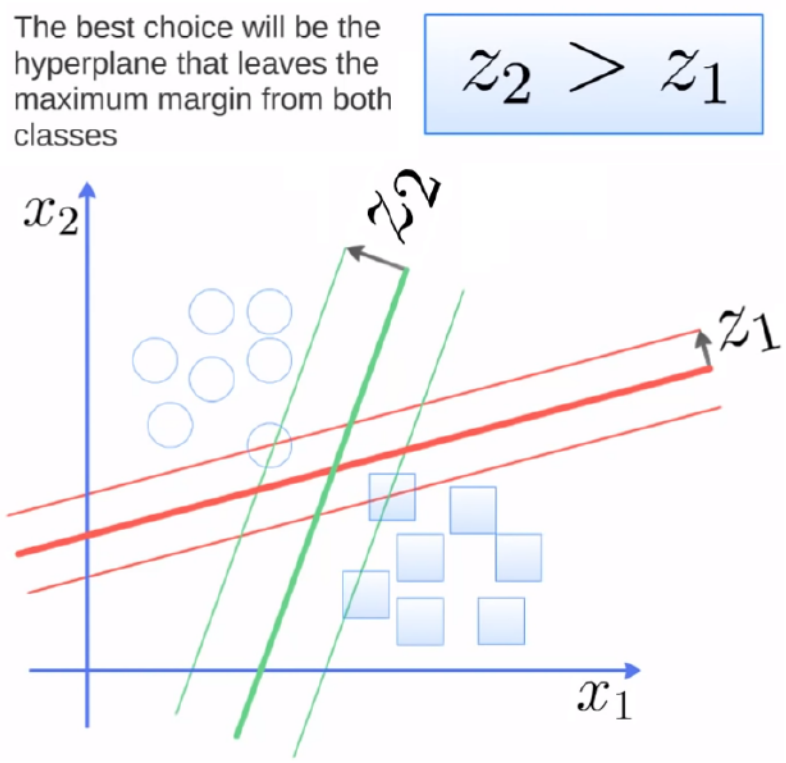


Figure 2.5: SVM hyperplane selection example

the Bidirectional Long Short-Term Memory (BiLSTM) approach presented by Graves and Schmidhuber in [24, 107].

2.3.2.2 Semi-Supervised or Minimally-Supervised Approaches

Semi-supervised approaches take a middle ground strategy by using a secondary small sense-annotated corpus as seed data, then applying a bootstrapping process such as the one presented in [25]. The bootstrapping technique requires only a small amount of tagged data that acts as seed data. This data then undergoes a supervised method to train an initial classifier, which is, in return, used on another untagged portion of the corpus to generate a larger training dataset. Only high-confidence classifications are considered as candidates for the final training dataset. Those same steps are then repeated in numerous iterations, and the training portion successively increases until the entire corpus is trained, or a maximum number of iterations caps the process. The main advantage of the bootstrapping approach is that it requires a small seed dataset to begin the training process.

The seed data could be manually-annotated or generated by a small number of surefire decision rules (e.g., the term bank in the context of a water body almost always indicates the riverside, the 1st noun sense from Fig. 1.3).

Under the same category, a semi-supervised approach is achieved by employing word-aligned bilingual corpora. This method assumes that an ambiguous word in one language is not ambiguous in another [108].

A more recent approach combined the original LSTM and Word2vec models from [106] and [54, 104], respectively. Then a semi-supervised algorithm annotates unlabeled sentences with those of similar ones from a smaller labeled dataset by using label propagation technique. This method relies on co-occurrence information between the tagged and un-tagged corpora [109].

2.3.2.3 Unsupervised Approach

Unlike the previous two categories, unsupervised approaches do not require the prior knowledge of the text; hence, no manual sense-annotated corpus is required. Nonetheless, most techniques in this category still require a training corpus for an unsupervised training task. Algorithms from this group have been further categorized into three groups: context clustering, co-occurrence graphs, and word clustering.

Context Clustering: Algorithms of this approach bear the underlying assumption that words of the same sense occur in similar contexts. Thus senses can be induced from corpus by clustering word occurrences with respect to their contexts. This is a clustering task over a context feature vectors for each word that appear in the training corpus. It is also referred to as Word Sense Induction (WSI) as introduced by Schütze [110, 111]. This group of algorithms follows three basic steps:

1. A context vector $\overrightarrow{c}$ is created for each instance of word w in the training corpus. The context vector could include features such as: Part Of Speech (POS), morphology, lemma, position, and other surrounding words.

2. Employ a clustering algorithm to cluster each instance into a predefined number of clusters, where each cluster represents a sense of the word w. Some of well known clustering algorithms used for the purpose of this task are: K-means clustering algorithm [26], agglomerative clustering algorithm [111, 112] etc.

3. Compute a cluster centroid vector $w_{\overrightarrow{s_j}}$ that represents a specific sense of word w.

The above steps represent the training part of the algorithm, as for the disambiguation of an instance term t of word w, a context vector $\overrightarrow{c}$ is created for the term t, then t is assigned to the closest sense (cluster) vectors $w_{\overrightarrow{s_j}}$ of w using a vector-based similarity such as cosine similarity [36].

Co-occurrence Graphs: A Different view of WSD is presented by the co-occurrence Graph method. In this technique, a co-occurrence Graph $G = (V, E)$ is constructed based on the co-occurrence of the syntactic relations (E) between words (V) within the same sentence, paragraph, document, or specified context. Dorow [113] and Widdows [114] presented such graph construction based on grammatical relations as follows:

1. For an ambiguous word w, a graph G_w is constructed with all words appearing in the context.

2. An adjacency matrix is determined for the G_w graph, and interpreted as a Markov chain.

3. Apply Markov clustering algorithm to determine word senses.

Other algorithms, based on the co-occurrence graph, have been further developed to enhance WSD task such as *HyperLex* [115], and *PageRank* WSD algorithm [116] which is based on the original *PageRank* algorithm developed by [117].

Word Clustering: In comparison to context clustering, this method aims to cluster semantically similar words that can convey a distinct meaning. One of the first models for word clustering was presented by Lin Dekang [118]. Given a target word w_t, the

model identifies semantically similar words (i.e. synonymous) based on their syntactical dependencies. The model constructs a similarity tree with a single node w_t. The tree branches expand based on similar words, where, by using pruning algorithm, each branch is differentiated as a distinct sense of w_t [118].

Clustering by Committee (CBC) is another technique presented by Lin and Pantel based on a similarity matrix of a given corpus. A clustering algorithm (i.e., average linkage clustering) is used to cluster similar words into groups called committees that each represents a specific sense. Finally, a target word is matched to one of the committees (senses) based on its similarity with the centroid [27].

2.3.2.4 Dictionary/knowledge-based Approaches

The main advantage of approaches of this category is that they do not require an intensive training process. However, they disambiguates words in context by exploiting large scale knowledge resources (i.e., dictionaries, ontologies, and KG). The most common techniques within this category, which are the focus of this study, are described in detail below.

Definition Overlap Systems: The definition overlap, or Lesk algorithm named after its author, is based on the commonality of words between two sentences, where the first sentence is the context of word w_t and the the second is the definition of a given sense from the knowledge base [29]. The definition with the highest word overlap is considered the correct sense. However, the Lesk algorithm has major limitations, i.e., being highly sensitive to the exact word match and having a concise definition within WordNet. To overcome this limitation, Nanerjee and Padersen [30] extended Lesk's algorithm to include related concepts within the knowledge base. Related concepts are identified through direct relations with the candidate sense (e.g., hypernyms or meronyms).

Semantic Similarity Systems: Since the introduction of WordNet, many semantic similarity and relatedness measures have been developed. Some of the most relevant mea-

sures were discussed in [9, 119]. This technique follows the intuition that words that appear in a sentence are coherently contextual, and should therefore be highly related within a conceptual knowledge base such as WordNet.

Soler and Montoyo proposed a verb WSD method based on the WordNet hierarchy. They first identified the verb-object phrase that needs to be disambiguated. Then, they extracted all nouns within the verb's gloss (definition) to determine its similarity with the following noun object in the phrase. Finally, a similarity matrix is constructed between nouns of each verb sense and the object senses. The similarity measure they used is based on the depth distance of the compared concepts and their LCS. For example, *"He is writing an article"*, the target phrase is *write-article*, and the nouns from the first sense *write-1* are {*student, thesis, week*}, Finally, the similarity matrix is constructed as shown in Table 2.5. The verb sense with the maximum total similarity is considered the correct sense (see Table 2.6) [1].

Table 2.5: Similarity matrix between write-1 and article senses [1]

write-1	atricle-1	atricle-2	atricle-3	atricle-4
student-1	0.31	0.37	0	0
student-2	0.45	0.40	0	0
thesis-1	0.67	0	0.70	0.40
thesis-2	0.72	0	0.94	0.44
week-1	0.29	0	0.30	0.30
week-2	0.29	0	0.30	0.30
week-2	0.26	0	0.27	0.27
Total	2.99	0.77	2.51	1.71

Table 2.6: Total matrices similarities between write and article senses [1]

Totals	atricle-1	atricle-2	atricle-3	atricle-4
write-1	**2.99**	0.77	2.51	1.71
write-2	2.84	0.83	2.45	1.46

Pedersen et al. [120] introduced a variation to the Lesk overlap approach by proposing an exhaustive evaluation of all possible combinations of sentences that can be constructed by all candidate senses within a context window. The context window is the words surrounding a target word. The Pedersen algorithm can be expressed as a general disambiguation framework based on a semantic similarity score. The framework can be described as follows: for a target word w_i, $\hat{S}$ is chosen such that it maximizes the sum of the most similar sense with all other words' senses based on the following equation [37, 120]:

$$\hat{S} = \underset{S \in Senses(w_i)}{\arg max} \sum_{w_j \in T : w_j \neq w_i}^{s} \underset{S' \in Senses(w_j)}{max} score(S, S'), \quad (1)$$

where $T = (w_1, ..., w_n)$ is the set of all words in a text, $Senses(w_i)$ is the full set of senses of $w_i \in T$. The formula measures the contributions of all context words with the most suitable sense. Pedersen's algorithm, as shown in Algorithm 1, can use any semantic similarity and relatedness measure. However, their results as reported in [120] are much lower than some of the recent approaches of this category, as shown below:

Algorithm 1: Maximum Relatedness Disambiguation [120]

Input : w_t: Target word

Output: i: Index of maximum related sense

```
foreach Sense s_ti ∈ Senses_of(w_t) do
    Initialize score_i ← 0
    foreach word w_j ∈ ContextWindow(w_i) = {w_j : j ≠ i} do
        Initialize maxScore_j ← 0
        foreach Sense s_jk ∈ Senses_of(w_j) do
            if maxScore_j < relatedness(s_ti, s_jk) then
                maxScore_j = relatedness(s_ti, s_jk)
        if maxScore_j > threshold then
            score_i+ = maxScore_j
Return i such that score_i ≥ score_j, ∀j, 1 ≤ j ≤ n, n = number of words in the
sentence.
```

A more recent study conducted by Mittal and Jain [121] utilized an average of three semantic similarity measures, some of which include Wu and Palmer (Sim_{wu}) measure [60], Leacock and Chodorow path-based measure (Sim_{lch}) [122], and a node counting distance measure. The average of all three similarity measures is assigned as a similarity value between each sense of an ambiguous word and all neighboring words (context) [121].

Heuristic Systems: Based on linguistic properties, heuristics are applied to evaluate word senses. The main idea is based on the ranking of sense distribution within a training dataset. Three main heuristic models have been developed to solve the WSD task: MFS, one sense per discourse, and one sense per collocation.

1. MFS is based on the frequency distribution of senses within the training dataset (i.e., SemCor and OMSTI). For a word w, the sense with the highest frequency is ranked first w_s^1, and the sense with the second highest frequency is ranked second w_s^2, and so on. Table 2.7 depicts the ranking of the noun senses for *'plant'* within SemCor dataset. In fact, senses in WordNet itself are ranked based on their frequency of occurrence in semantic concordance texts[6] [37].

2. One sense per discourse argues that the meaning of a word is most likely preserved within a specific text/domain, rather than in general.

3. One sense per collocation narrows the preservation of meaning within collocation instead of a domain.

Once the challenging part of ranking the senses within the knowledge base is completed, disambiguating a word would be as simple as selecting the most frequent sense from the training dataset; which is referred to as MFS baseline. The first sense selection from WordNet is also considered a baseline approach. These baseline approaches yield a moderate accuracy between 55.2% and 67.8% as reported in SemEval-07 and SemEval-15, respectively [123].

[6] `https://wordnet.princeton.edu/documentation/wndb5wn`

Table 2.7: WordNet sense ranking based on SemCor frequencies

Sense	**Definition**	**Frequency**
plant-1	Buildings for carrying on industrial labor	338
plant-2	A living organism lacking the power of locomotion	207
plant-3	Something planted secretly for discovery by another	2
plant-4	An actor situated in the audience whose acting is rehearsed but seems spontaneous to the audience	0

Graph-based Systems: Several other methods exploited the knowledge base structure and attempted to construct a sub-graph to determine the appropriate sense within a sentence. Navigli and Lapata constructed a graph containing all possible combinations of the ambiguous words' senses. Where each node of the new graph represents a sense of one of the word sequence, while edges correspond to relationships between senses. Once the graph is constructed, each node is assessed based on the shortest path measure to determine the most suitable sense for each word that provides the highest context [124].

The final literature of this technique was presented by Dongsuk et al. [125] that proposed a new vector-based semantic similarity measure and an iterated WSD technique. Their semantic similarity measure is based on the construction of a sub-graph for each sense, then a semantic relational path is extracted using Depth First Search (DFS) algorithm. Finally, by regarding relationships as words, and semantic relational path as sentences, they implemented an unsupervised learning approach (i.e., Doc2Vec) to generate a vector for each sentence [126]. Sentences with similar semantic relational paths are projected to a similar vector space. The semantic similarity measure is then calculated using a standard cosine similarity [125].

Out of all WSD approaches, this thesis focuses on the knowledge-based approaches, due to their ability to exploit massive semantic knowledge graphs. Another advantage of knowledge-based systems is their independence of expensive sense-tagged corpus, which led to rapid developments of such systems in recent years. Finally, the recently developed knowledge-based systems narrowed the performance gap with their peers of the super-

vised systems, and in some cases they have outperformed them.

2.3.3 State-of-the-art Knowledge-based Systems

The following are the knowledge-based systems that have been used as a benchmark and are compared to our system.

Lesk: The original Lesk algorithm is based on a gloss overlap between the definitions of the ambiguous word's senses and its sentence (i.e., context). The sense definition with the maximum overlap with the word's sentence is selected as the correct sense [29]. Lesk_{ext} is an extension of the original gloss overlap, which extended the gloss to include terms that share one or more relations with the ambiguous term in the KG. They also employed the TF-IDF weights to compute the final similarity between the extended gloss and the context [30]. Finally, $\text{Lesk}_{\text{ext+emb}}$ incorporated Latent Semantic Analysis (LSA) to select the appropriate sense using semantic vector similarity instead of TF-IDF vector similarity. They re-weighted the terms using an Inverse Glass Frequency (IGF), viewing all extended glosses as a corpus compared to the Inverse Document Frequency (IDF) approach. Beyond using the distributional semantic space, the latter overcame the bag of words overlap limitation in the original Lesk algorithm by using a vector cosine similarity [127]. However, the Lesk algorithm is dependent on the matching of terms between the compared texts. Moreover, the algorithm would fail if the compared text contains synonym terms rather than the exact terms. In addition, none of the overlap approaches take into consideration the sequence of terms within the sentence itself.

UKB: UKB employed a graph-based PageRank approach on the entire WordNet graph, which is a completely different approach from Lesk's. To optimize the PageRank algorithm over WordNet, they constructed a subgraph for a text window (typically a sentence or few contiguous sentences). The subgraph included the senses of all open-class (ambiguous) terms and the rest of the text as a context [28]. An extended version of UKB, namely $\text{UKB}_{\text{gloss}}$, employed extended WordNet to transform the glosses into dis-

ambiguated synsets. This implementation of UKB also incorporated sense frequencies to initialize context words [128]. The latest release of UKB is $\text{UKB}_{\text{gloss18}}$, which includes the optimal parameters for the software to guarantee optimal performance. For example, they used over 20 words window as a context of each target word and 30 iterations for the personalized PageRank algorithm. They also confirmed that using WordNet versions 1.7.1 and 2.0 resulted in better performance since they match the annotated datasets [129]. Furthermore, the authors highlighted the use of an undirected graph as a limitation for the PageRank algorithm [128].

Babelfy: A graph-based approach integrated entity linking and WSD that is based on random walks with restart algorithm [130] over BabelNet, which is an extensive multi-graph semantic network integrating entities from WordNet and Wikipedia or Wiktionary. Babelfy employs the densest subgraph heuristic for selecting the most suitable sense of each text fragment. For a target word, Babelfy considers the entire document instead of the sentence alone [131]. This approach is also bound by the PageRank algorithm limitations with respect to WordNet KG.

WSD-TM: This is a graph-based WSD system that uses a topic modeling approach based on a variation of the Latent Dirichlet Allocation (LDA) algorithm. This approach employes the whole document as a context to disambiguate all open-class words within the document. WSD-TM views document as synsets and synset words rather than topics and topic words, then performs the LDA algorithm based on that assumption [132].

Baselines: Senses in WordNet are ranked based on their frequency of occurrence in semantic concordance texts[7]. Therefore, selecting the first sense of the target word in WordNet is presented as a baseline. Another baseline is based on the MFS extracted from the training dataset (SemCor and/or OMSTI).

To summarize, Table 2.8 highlights the main characteristics of the benchmark systems.

[7] `https://wordnet.princeton.edu/documentation/wndb5wn`

Table 2.8: knowledge-based WSD system

System	Algorithm	Similarity Measures	KG
Lesk [29]	Definition overlap	Lesk	WordNet
Lesk_{ext} [30]	Definition overlap	Lesk	WordNet
$\text{Lesk}_{\text{ext+emb}}$ [127]	Definition overlap	Lesk	WordNet
UKB [28]	PageRank	JCN, LCH, LESK	WordNet
$\text{UKB}_{\text{gloss}}$ [128]	PageRank	JCN, LCH, LESK	WordNet
$\text{UKB}_{\text{gloss18}}$ [129]	PageRank	JCN, LCH, LESK	WordNet
Babelfy [131]	PageRank	Undefined	BabelNet
WSD-TM [132]	LDA	LESK	Wiki
WN1^{st} sense	1^{st} Sense	NA	WordNet
MFS_{s}	Heuristic	NA	NA

2.3.4 Critical Analysis of the Related Work

Although the above-mentioned benchmark systems are all knowledge-based, they can be further classified into three subcategories based on their implemented algorithm. The first subcategory is the definition overlap, the second is the graph-based (i.e., PageRank), while the third is topic modeling. The Lesk systems follow the definition overlap, which limit the similarity between two texts on the term's exact match. Furthermore, the original Lesk algorithm adopts a bag-of-words approach. This was enhanced with a vector-based in subsequent literature. However, none of the overlap methods considered the broader context of the document.

The UKB systems employ a graph-based method (i.e., PageRank). The PageRank algorithm is time-consuming and requires intensive computational power to weigh the links

between WordNet concepts. Furthermore, some of these systems employ the Lesk algorithm for the initial weights linking any two concepts [133], while others use a collection of semantic similarity measures including JCN, LCH, and Lesk [128, 134]. The personalized PageRank optimizes the performance by using a subgraph approach. However, this is done at the cost of context reduction, as the optimal results of UKB considers a window size of 20 words, which could span multiple sentences [129].

The WSD-TM system relies on the document topic as the main disambiguating context. Despite the importance of the global document context, the WSD-TM overlooks the importance of the word's local surroundings, which is considered a local context. Furthermore, this system also employs Lesk similarity to model relationships between synsets as one of its priors to the LDA algorithm. A major limitation that applies to most systems in these three categories is that they follow a bag of word approach, ignoring the sequence of the terms within the sentence, which we believe is a critical factor to disambiguate a word within its sentence and discourse contexts.

Research published in neuroscience journals shows that human brain models suggest that semantic memory is a construction of the conceptual knowledge based on a widely distributed network [135]. Based on some models, the brain networks consist of neurons, neuronal populations, or brain regions that can be viewed as nodes, and the structural or functional connectivity viewed as edges linking these nodes together [136]. Fig. 2.6 describes such a network with functional relationships connecting various brain regions (nodes). Furthermore, structural or functional connectivities refer to the anatomical pathways between neurons, neuronal populations, or brain regions, depending on the spatial scales of interest. These structural and functional connections form a biological route for information transfer and communication [135, 137]. If we compare the KG to our brain, viewing concepts as nodes and relations as structural and functional connections, we can rely on widely distributed KG to extract various semantic knowledge, including similarity and relatedness between nodes using the structural and functional relationships, respectively.

Inspired by the brain models, we try to overcome the limitations mentioned above as

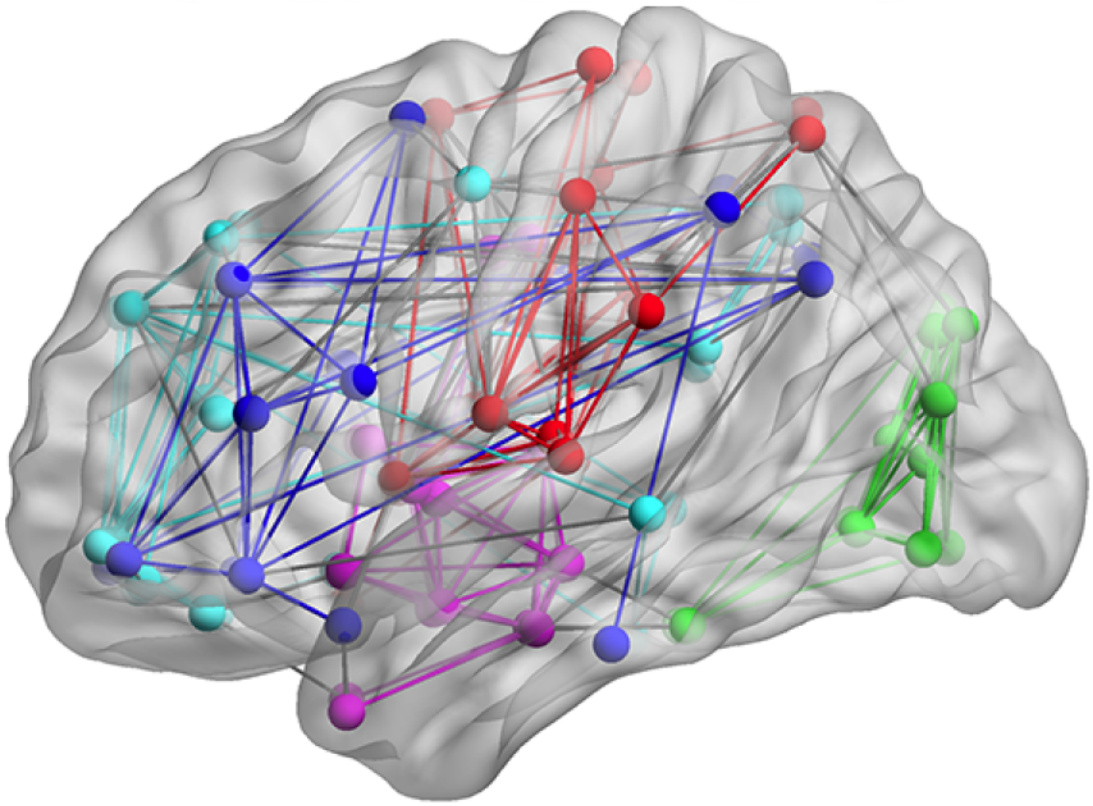


Figure 2.6: Visualization of the human brain network using the BrainNet viewer [138]

follows: we argue that the sequential connectivity of terms has an essential part in forming the overall context of the sentence. Beneath the sequential connectivity, there exists structural and functional relationships that construct the term's context. These relationships are measured by semantic similarity and relatedness within the KG.

Consider the following two sentences:

- *"John has all his **faculty** members at the meeting table."*
- *"John has all his **faculties** and could think clearly and logically"*

The word *faculty* (lemma of *faculties*) has two distinct meanings (see Fig. 2.7), and without the rest of the sentence or other external context (e.g., knowing that John is a dean at a university), it is challenging to distinguish the correct meaning. Since humans use and rely on context to disambiguate words, machines are even more dependent on it.

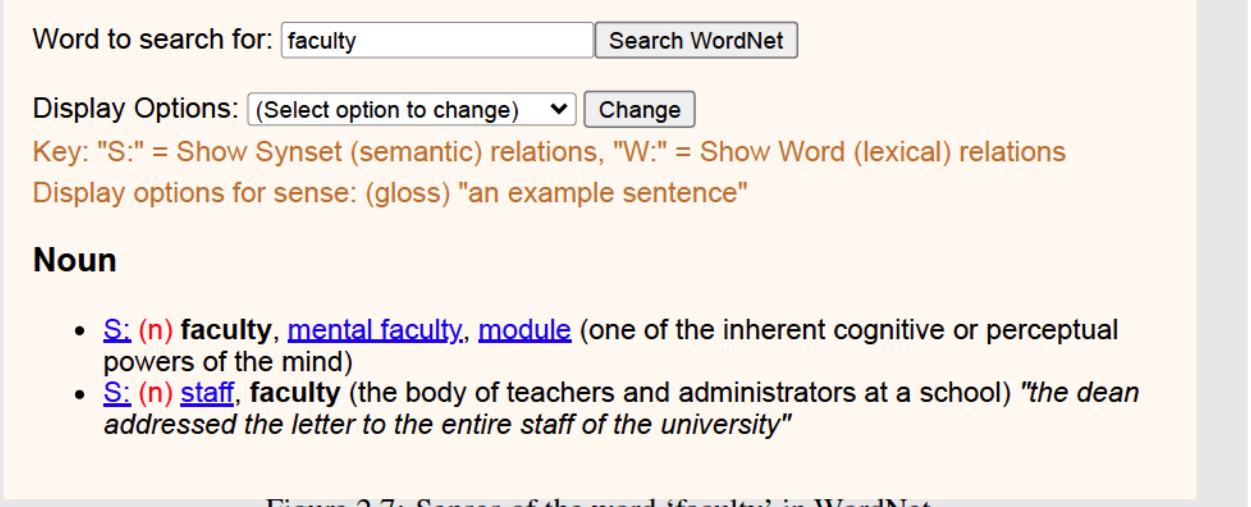


Figure 2.7: Senses of the word 'faculty' in WordNet

If we remove the terms after *faculty* from both sentences, it will not be easy, as a human being, to understand the correct meaning. This difficulty is derived from the fact that the term *faculty* is ambiguous. However, as we add more context to the sentence, the meaning becomes more evident in each sentence. More importantly, our brain will be able to establish functional connectivities between the terms of the sentence and infer additional knowledge, such as *John* could be working at a university as a chair or a dean.

Initially, our brain could not understand the meaning of *faculties* because it could not make the connection between the term and its surrounding context {'John', 'has', 'all', 'his'}. However, as soon as the context is enriched with {'members', 'at', 'the', 'meeting', 'table'}, our brain was able to create a context from the joint meanings of the core terms in the sentence {'John', 'faculty', 'member', 'meeting', 'table'}, hence, disambiguate the sentence. Surprisingly enough, the three terms {'member', 'meeting', 'table'} are also ambiguous terms, with even more senses to choose from, see Fig. 2.8. However, our brains can connect the various meanings of each term and determine the context of the full sentence. Our main observation here shows us that humans tend to connect terms/things based on the various associations that connect them, in addition to its prior heuristic knowledge about the ambiguous terms. The prior heuristic knowledge is represented by the common use of the terms presented in the sequence.

To summarize, the four point below are essential for disambiguating words within a sentence, hence, we incorporate them into our proposed WSD algorithm:

- The sequence of the terms within the sentence
- The connectivity between various concepts (i.e., senses) of ambiguous terms.

- A basic heuristic knowledge of each term and its various concepts (i.e., senses)
- The broader context of the document.

2.4 Conclusion

Various semantic similarity and relatedness measures have been proposed in the literature, most of which rely on a single semantic relation: the ISA taxonomic relation. Few measures utilized the part-of or antonym non-taxonomic relations within WordNet. However, limiting these methods to one or two non-taxonomic relations within WordNet makes them rigid and limited in evaluating semantic similarity and relatedness between terms. To our knowledge, none have presented a comprehensive method that is adaptable to other relations that exist in a domain-specific KG. Semantic similarity and relatedness measures have shown limited performance and computational complexity when solving knowledge-based WSD tasks.

[8] `http://wordnetweb.princeton.edu/perl/webwn`

WordNet Search - 3.1
- WordNet home page - Glossary - Help

Word to search for: member Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- S: (n) **member**, fellow member (one of the persons who compose a social group (especially individuals who have joined and participate in a group organization))

WordNet Search - 3.1
- WordNet home page - Glossary - Help

Word to search for: meeting Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- S: (n) **meeting**, group meeting (a formally arranged gathering) *"next year the meeting will be in Chicago"; "the meeting elected a chairperson"*
- S: (n) **meeting**, get together (a small informal social gathering) *"there was an informal meeting in my living room"*

WordNet Search - 3.1
- WordNet home page - Glossary - Help

Word to search for: table Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- S: (n) **table**, tabular array (a set of data arranged in rows and columns) *"see table 1"*
- S: (n) **table** (a piece of furniture having a smooth flat top that is usually supported by one or more vertical legs) *"it was a sturdy table"*
- S: (n) **table** (a piece of furniture with tableware for a meal laid out on it) *"I reserved a table at my favorite restaurant"*
- S: (n) mesa, **table** (flat tableland with steep edges) *"the tribe was relatively safe on the mesa but they had to descend into the valley for water"*
- S: (n) **table** (a company of people assembled at a table for a meal or game) *"he entertained the whole table with his witty remarks"*
- S: (n) board, **table** (food or meals in general) *"she sets a fine table"; "room and board"*

Verb

- S: (v) postpone, prorogue, hold over, put over, **table**, shelve, set back, defer, remit, put off (hold back to a later time) *"let's postpone the exam"*
- S: (v) **table**, tabularize, tabularise, tabulate (arrange or enter in tabular form)

(c) Senses of the word 'table' in WordNet

Figure 2.8: Definitions for the terms walk and bank in WordNet[8]

Chapter 3

Poly-Relational Semantic Similarity and Relatedness Measure

3.1 Introduction

The rapid expansion of LOD requires a comprehensive semantic similarity measure that is yet to exist. DBpedia [38], Freebase [39], YAGO [40], and WordNet [42] are examples of lexical KG repositories that resemble LOD. KGs are representative of an ontological schema, which semantically models a specific domain of knowledge. In technical terms, an ontology is a formal semantic representation of the concepts within a specific domain. The semantic representation is established through a set of axioms. An axiom connects two concepts and/or instances through a specific relation that models real-world relationship in the form of *subject*, *predicate*, and *object*. An interconnected set of axioms forms a KG, or SG as referred to in some literature [3, 4, 5].

This chapter presents the first cornerstone of this thesis, a Poly-Relational Semantic Similarity and Relatedness (PR-SSR) measure that comprises all semantic relations between concepts. The rest of the chapter is organized as follow: section 3.2 describes the architectural design for the PR-SSR. Section 3.3 presents the proposed parameters for each relation which are used in the new measure presented in the next Section 3.4. Fi-

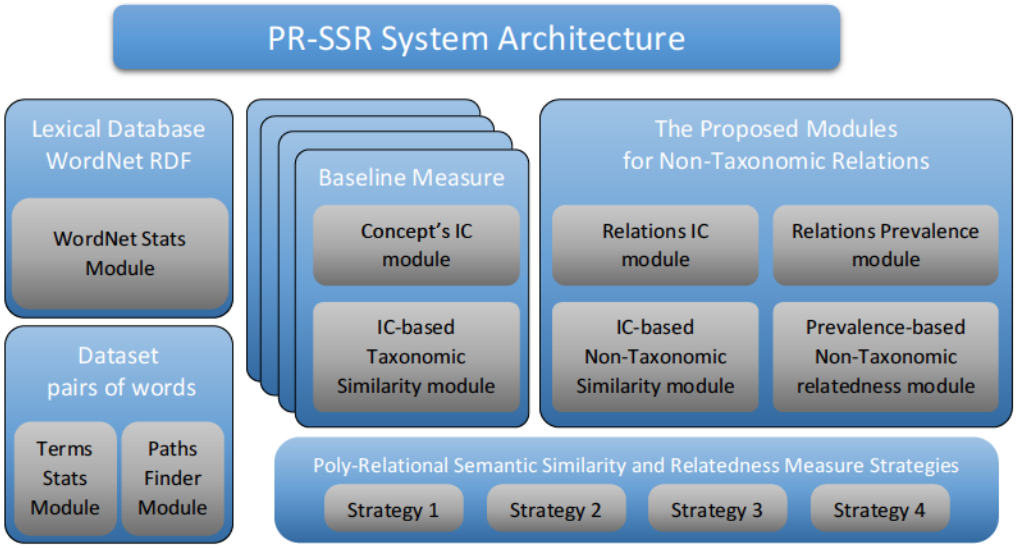


Figure 3.1: PR-SSR system architecture

nally, and before we conclude the chapter, Section 3.5 presents the experiment's setup, evaluation metrics, implementation, and the experimental results.

3.2 System Architecture

Fig. 3.1 shows the PR-SSR system architecture, which includes three offline statistical modules: the WordNet stats, term stat, and paths finder modules. These modules pre-calculate the required statistics for the WordNet Resource Description Framework (RDF) graph and the test dataset when performing testing on a pre-existing gold standard dataset, such as the ones described below in Section 3.5.1.1. The architecture also includes the implementation of six existing baseline measures; these are the same measures that are described in Section 2.2.2. We measure our system based on the improvement of these baselines. The core of our system consists of four components, Relations IC, relations prevalence, IC-based non-taxonomic similarity (we refer to it as $RelSim$), and prevalence-based non-taxonomic relatedness modules. These components implement our proposed Semantic Information Content (SemIC), prevalence, $RelSim$, and relatedness, respectively. Finally, the last component is the weighted combination of the baseline

taxonomic similarity $TexSim$, the proposed relational-based similarity $RelSim$, and the proposed non-taxonomic relatedness in one of the four strategies.

3.2.1 System Flowchart

Fig. 3.2 depicts the flowchart for calculating the semantic similarity and relatedness. Unlike other methods, we combine existing taxonomic measures with the proposed non-taxonomic relational-based similarity and non-taxonomic relatedness measures between terms.

The process starts after extracting the compared terms statistics. These statistics include the LCS for the compared terms, also the number and types of all non-taxonomic relations. We use WordNet v3.1 to calculate these statistics. The next step starts by calculating the taxonomic IC for each term and then the semantic similarity using the calculated IC. Using the concept's IC, we calculate a local IC for each relation Relation Information Content (RIC). Having the relation-IC, we then compute the proposed relational-based similarity ($RelSim$) between terms. In Section 3.4.1 we present three strategies to compute $RelSim$. If there exists any non-taxonomic path between the terms, we calculate the non-taxonomic relatedness between the terms as an edge-weighted path using the relations distribution within the KG (prevalence) as discussed in Section 3.3.2.

3.3 New Semantic Similarity and Relatedness Parameters

3.3.1 Relation IC

Similar to concepts, relations are organized into hierarchical structure within their ontology. For instance, the Wordnet ontology presented earlier in Fig. 2.1a shows a sub-set of the relations' hierarchy used in Wordnet, where *topObjectProperty* is the most general relation, and *Hypernym, Hyponym, and others* are some of the most specific relations in

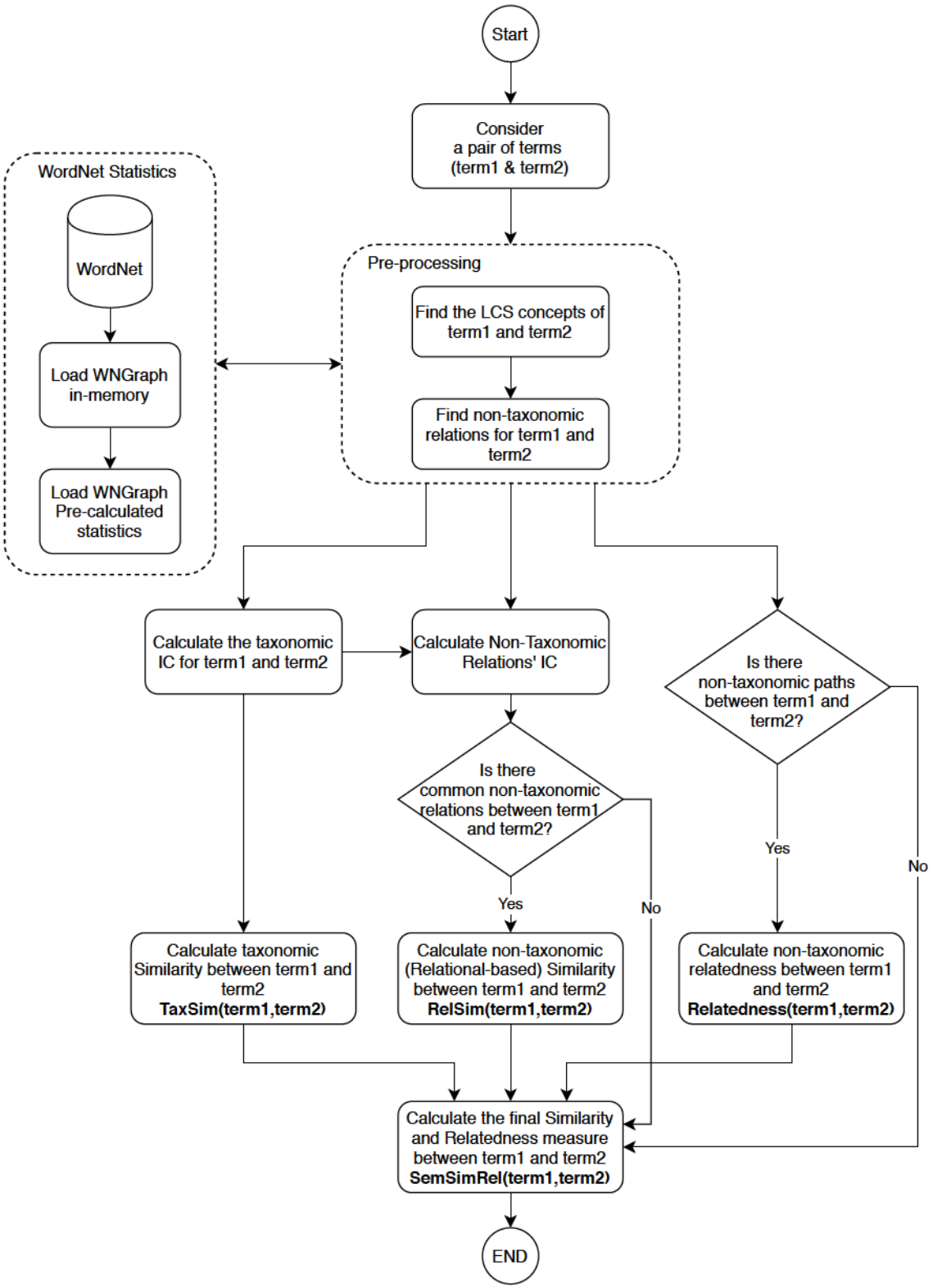


Figure 3.2: Flowchart for the semantic similarity and relatedness algorithm

the hierarchy. Based on such hierarchy, concepts gained an intrinsic IC attribute, and likewise, do relations. Having this intuition, we utilize existing IC-intrinsic measure with relations to compute the $IC_{Tax}(r_t)$ as the taxonomic-based IC for relation r of type t. Eq. (1) simply reflects the taxonomic IC for each relation type.

$$IC_{Tax}(r_t) = ic_{base}(r_t), \tag{1}$$

where ic_{base} denotes the baseline taxonomic IC measure from Table 2.1.

Furthermore, domain and/or range concepts in an ontology may contextualize a relation. The IC of domain and range concepts enrich the associated relation as global contextual information. Intuitively, we argue that relations assimilate global contextual information from their domain and range concepts, which is proportional to their depths. Therefore, we define the global IC for a relation as a monotonically increasing function with respect to both the IC and depth of the domain and range concepts, as shown in Eq. (2). Consequently, for two pairs of concepts with the same absolute IC difference, the deeper the pair, the greater the global IC. Also, for two pairs of concepts at the same depth, the greater absolute IC difference, the greater the global IC. It should be noted here that Eq. (2) is a monotonically increasing function with respect to both the IC and depth of the domain and range concepts, as shown below:

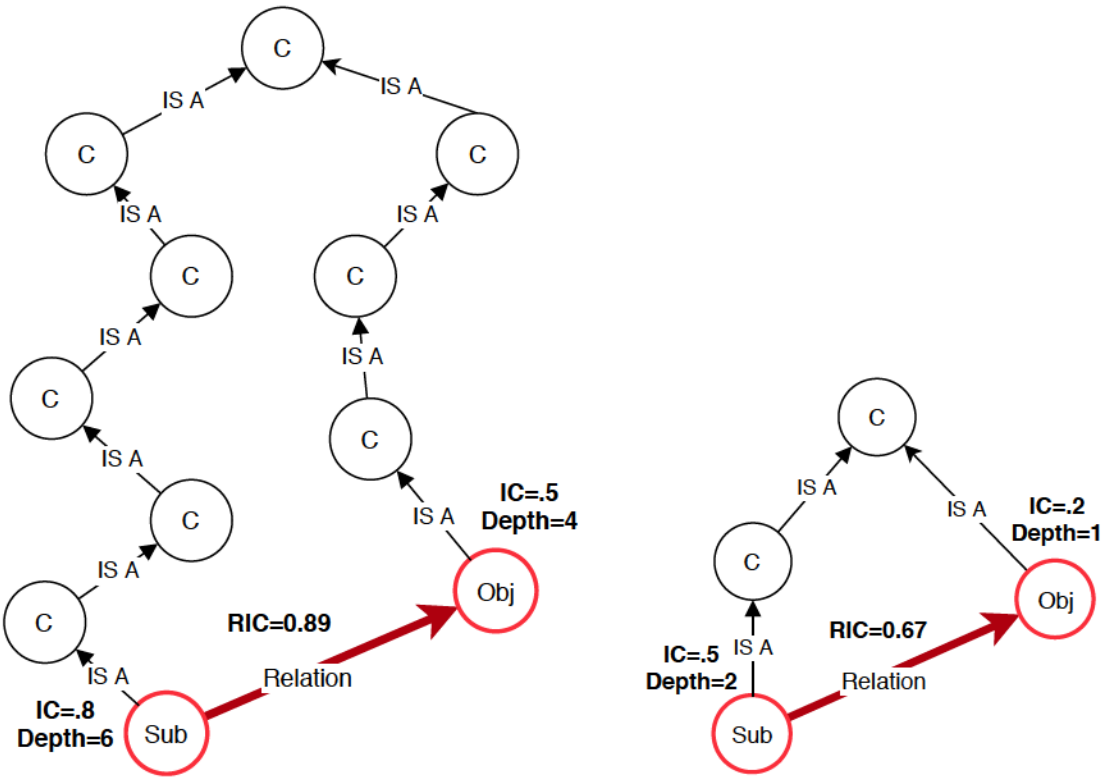
$$IC_{GC}(r_t) = \left| ic_{base}\Big(Dom(r_t)\Big) - ic_{base}\Big(Ran(r_t)\Big) \right|^{\Psi}, \tag{2}$$

where $ic_{base}(Dom(r_t))$, and $ic_{base}(Ran(r_t))$ are the ICs of the domain and range concepts respectively, and $\Psi = [1/(deep(Dom(r_t)) + deep(Ran(r_t)))]$, where $deep$ is the depth function.

Similar to the global context, an instance relation attains information from its subject and object within the KG. Hence, local contextual information is assigned to each instance relation based on its subject and object. The local IC for a relation instance of type t is defined in the equation below:

$$IC_{LC}(r_t) = \left| ic_{base}\big(Sub(r_t)\big) - ic_{base}\big(Obj(r_t)\big) \right|^{\Upsilon}, \tag{3}$$

where $Sub(r_t)$, and $Obj(r_t)$ are the subject and object concepts, respectively, and $\Upsilon = [1/(deep(Sub(r_t)) + deep(Obj(r_t)))]$. Fig. 3.3 illustrates the effect of the depth of the



(a) Deep concepts (b) Shallow concepts

Figure 3.3: Concept's depth effect on RIC

concepts on the relations' IC values. This effect applies to both the global and the local ICs within the ontological schema and knowledge graph instances respectively. Fig. 3.3a shows a RIC between deeper concepts while Fig. 3.3b illustrates an RIC between shallow concepts.

Finally, the taxonomic, global context, and local context IC values are combined to form the RIC, as shown in the following equation:

$$RIC(r_t) = \alpha \times IC_{Tax}(r_t) + \beta \times IC_{GC}(r_t) + \gamma \times IC_{LC}(r_t), \tag{4}$$

where α, β, and γ are constants to measure the contribution of each IC function. These constants are contextually selected based on the actual conceptual schema and hierarchical structure of the ontology and KG describing the modeled knowledge.

3.3.2 Relation Prevalence

Based on the second observation described in Section 1.2, relations exist in a KG in accordance to their relevancy to the modeled domain. Consequently, important relations are used more frequent than other less important relations, because they relate more to the modeled domain. For example, a *father-of* and *mother-of* relations are more prevalent in a Family-domain KG than a Plant-domain KG. Therefore, the probability of a relation in a KG, Eq. (5), measures the contribution of that relation to the semantic information added to each associated concept.

$$P(r_t) = \frac{freq(r_t)}{Total\ number\ of\ relations}, \tag{5}$$

where $freq(r_t)$ is the total number of relations of type t.

3.3.3 Poly-Relational similarity and relatedness measure

The relations' IC and prevalence are the core ingredients for the proposed relational-based similarity and relatedness measure, respectively. The two proposed measures complement the existing taxonomic-based similarity measures, and together they form a comprehensive poly-relational semantic similarity and relatedness measure. Therefore, we propose a semantic similarity and relatedness measure between two terms as a function of their taxonomic similarity, relational similarity, and relatedness.

$$SemSimRel(w_1, w_2) = f\Big(TaxSim(w_1, w_2), RelSim(w_1, w_2), Relatedness(w_1, w_2)\Big), \tag{6}$$

where $TaxSim(w_1, w_2)$ represents a general taxonomic IC-based similarity measure between w_1 and w_2 as described in Table 2.2, $RelSim(w_1, w_2)$ is the proposed relational-based non-taxonomic similarity presented in Section 3.4.1, and $Relatedness(w_1, w_2)$ is the proposed relatedness based on weighted non-taxonomic relational paths presented in Section 3.4.2.

in the next section we show how the relation's IC and prevalence are employed to devise a novel comprehensive semantic similarity and relatedness measure named PR-SSR.

3.4 Proposed Method

In this section we present four different strategies for our proposed PR-SSR Measure using the relation's IC and prevalence metrics presented in the previous section 3.3. The first three strategies exploit the relation's IC to compute a relational-based (non-taxonomic) similarity, while the forth strategy uses the relation's prevalence to compute the non-taxonomic relatedness.

3.4.1 Relational-based Similarity

In relational-based similarity, terms are considered similar based on the similarity of their semantic non-taxonomic relations, which could be viewed as functionality or attributes in some domains. For instance, In the family KG Fig. 1.1 'Meredith' and 'Ruth' have some similarity for being both mothers by having the same 'is mother of' relation. To demonstrate the benefit of employing relations to enhance semantic similarity, we propose three strategies showing their impact at different granularity levels. The first strategy makes use of all relations, regardless of their type, to compute a single semantic attribute. While, the second strategy exploits relations' types in computing similarity. Finally, the third strategy computes similarity based on instances of each relation type. This coarse-to-fine grain investigation provides new insights about the role of non-taxonomic relations in measuring semantic similarity. Furthermore, a fourth strategy, employing relations' prevalence, proposes a weighted non-taxonomic relational path to bring into perspective the role of relatedness in further enhancing semantic similarity.

Strategy 1

This strategy is based on the intuition of the original taxonomic IC, where each concept is attributed an IC value as a measure of various hierarchical features (i.e., hyponyms, depth, hypernyms, leaves, and siblings) as shown in Table 2.1. Similarly, we propose an additional SemIC attribute for each term, based on all associated non-taxonomic relations.

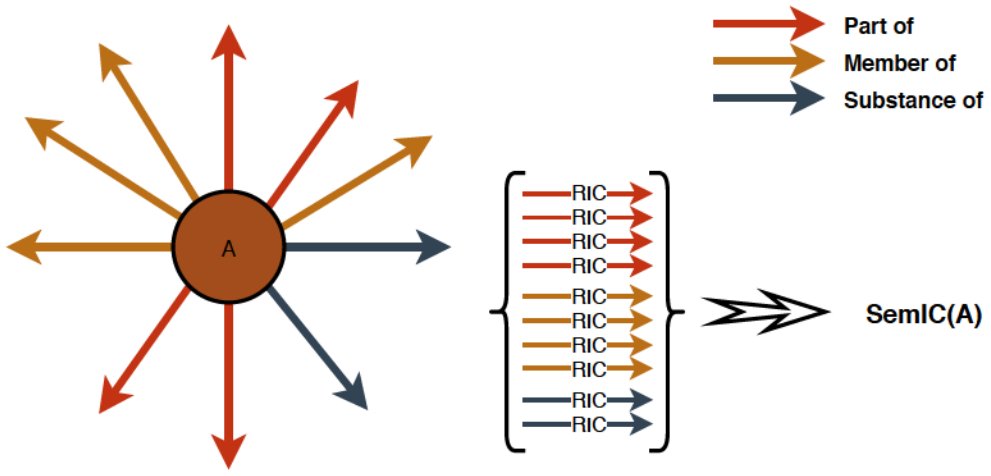


Figure 3.4: SemIC for concept based on strategy 1

This SemIC metric, defined in Eq. (7), is an aggregation of the RIC contribution of all associated relations.

$$SemIC(w) = log\left(\sum_{r_t \in RelSet:\left(w \xrightarrow{r_t} o\right)} (P(r_t) \times RIC(r_t)) + 1\right), \tag{7}$$

where $RelSet(w \xrightarrow{r_t} o)$ is the set of all relation instances that link term w with all other object terms. Fig. 3.4 illustrates a concept SemIC as a single value computed using Eq. 7.

The SemIC is then used to evaluate the semantic similarity between terms, using existing baseline similarity measures from Table 2.2. The relational similarity $RelSim(w_1, w_2)$ between w_1 and w_2 is computed by replacing the $ic(w)$ by $SemIC(w)$, as denoted below:

$$RelSim(w_1, w_2) = Sim(w_{1_{SemIC}}, w_{2_{SemIC}}), \tag{8}$$

where $w_{i_{SemIC}}$ refers to the semantic IC of term i instead of its taxonomic IC.

Strategy 2

The non-taxonomic semantic IC, SemIC, proposed in the first strategy assigns a non-discriminating semantic attribute to each concept. Although this attribute describes the

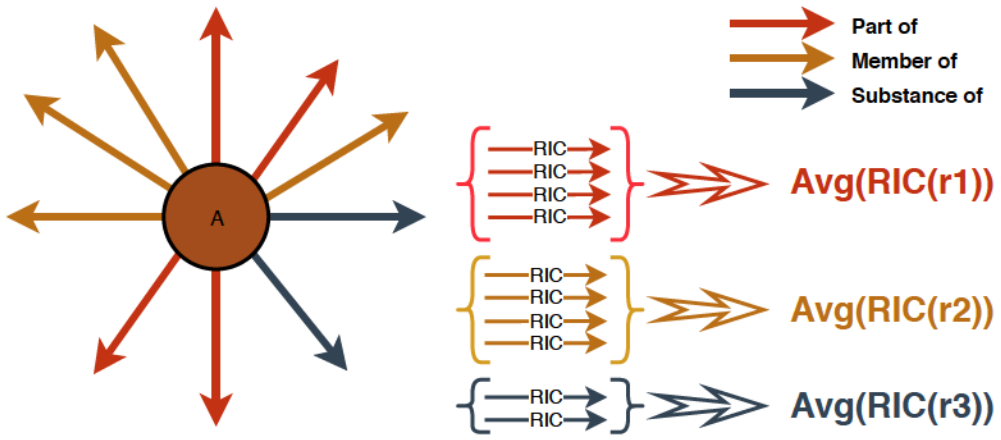


Figure 3.5: SemIC for concept based on strategy 2

semantic information contained within the concepts, yet, relations' types are being ignored in the similarity evaluation between two concepts. Furthermore, in Section 2.2.3, we demonstrated the importance of each relation type and its own implication on the semantic definition and IC of a concept. Therefore, the second strategy refines the first one by attributing a vector of RICs for each term. Each element of the vector represents the average RIC of a specific relation type. Fig. 3.5 illustrates three non-taxonomic relation types (part of, member of, and substance of). For each relation type, the average RIC is computed and used to build a final SemIC vector as shown in Eq. 9 below:

$$SemIC(w) = \vec{r}_w = \begin{bmatrix} Avg(RIC(r_{t1})) \\ ... \\ Avg(RIC(r_{tn})) \end{bmatrix}, \tag{9}$$

where $r_{ti} \in RTSet(w)$ denoting the set of relation types linking the term w to all other objects.

We then employ one of the existing vector-based similarity/distance measures, such as Euclidean, Hamming, Cosine, Mean-Squared Error (MSE), and Summation of Squared Difference (SSD), to measure relational similarity. We compared eleven distance measures from Math.NET numerics library[1].

$$MSE(\vec{x}, \vec{y}) = \frac{1}{n} \sum_{i=1}^{n} (x_i - y_i)^2, \tag{10}$$

[1] https://numerics.mathdotnet.com/Distance.html

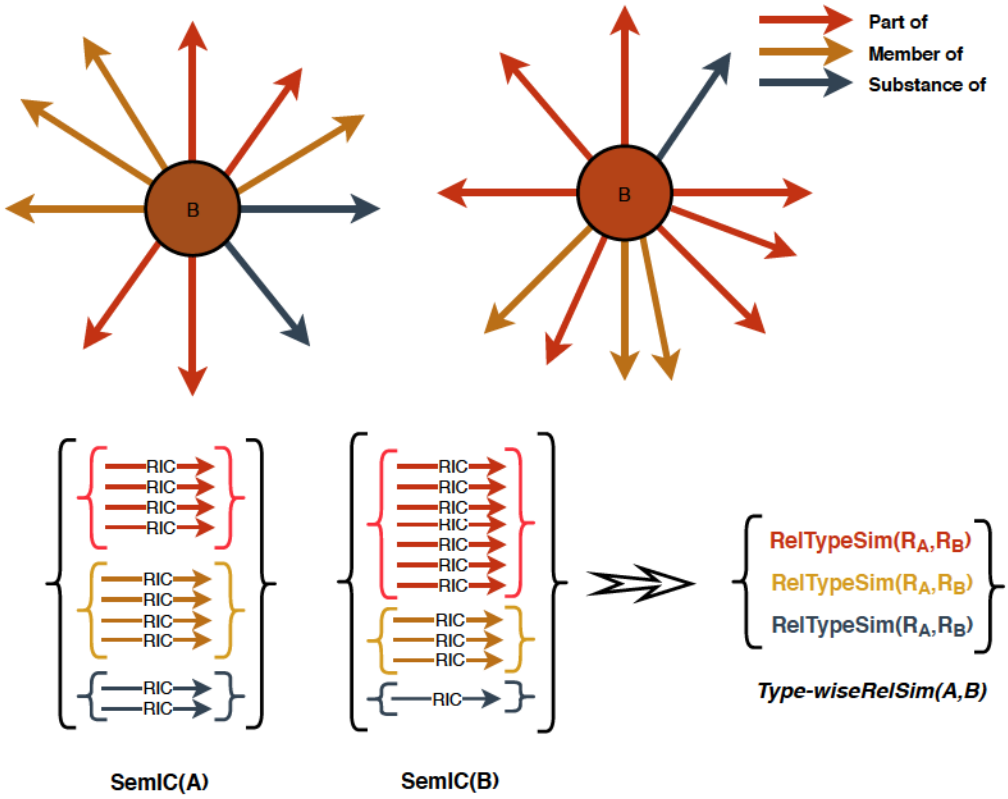


Figure 3.6: SemIC for concept based on strategy 3

The MSE distance, described in Eq. (10), provided the highest correlation results with the gold standard, and consequently, is used to compute the relational similarity as shown in Eq. (11)

$$RelSim(w_1, w_2) = 1 - MSE(\vec{r}_{w_1}, \vec{r}_{w_2}), \tag{11}$$

where $\vec{r}_{w_1}$ and $\vec{r}_{w_2}$ are the average relational type vectors of w_1 and w_2, respectively.

Strategy 3

This strategy is more granular than the previous ones, where we focus not only on the types of relations as we did in strategy 2, but also on each instance within each relation type. More specifically, this approach exploits the benefits of measuring similarity between instances of the common relation types of both terms. Then, aggregating the resulting type-wise relations' similarities to compute the overall relational similarity.

As illustrated in Fig. 3.6, first, for each relation type in the $RTSet(w)$, we generate a vector of RICs associated to the relation instances of that type. Then, these vectors are used to build a semantic IC of a term as a set of type-wise RIC vectors denoted by $\{\vec{r}_{ti}, i = 1..n\}$, where n is the number of relation types associated to that term, as shown in Eq. (12).

$$SemIC(w) = \left\{ \begin{array}{c} \vec{r}_{t1} = \begin{bmatrix} RIC(r_{t11}) \\ ... \\ RIC(r_{t1m_1}) \end{bmatrix} \\ ... \\ \vec{r}_{tn} = \begin{bmatrix} RIC(r_{tn1}) \\ ... \\ RIC(r_{tnm_n}) \end{bmatrix} \end{array} \right\}, \tag{12}$$

where m_i is the cardinality of instances of the i^{th} relation type. To compute the relational type-wise similarity between two terms, we use the MSE distance method as shown below:

$$RelTypeSim(r_{t_{w_1}}, r_{t_{w_2}}) = 1 - MSE(\vec{r}_{t_{w_1}}, \vec{r}_{t_{w_2}}) \tag{13}$$

Consequently, a type-wise relational similarity set for the common relation types between two terms can be expressed as follows:

$$Type\text{-}wiseRelSim(w_1, w_2) = \left\{ \begin{array}{c} RelTypeSim(\vec{r}_{t1_{w_1}}, \vec{r}_{t1_{w_2}}) \\ ... \\ RelTypeSim(\vec{r}_{tn_{w_1}}, \vec{r}_{tn_{w_2}}) \end{array} \right\} \tag{14}$$

Finally, the overall relational similarity between two terms is defined as the normalized sum of the weighted type-wise relational similarities. The normalization is based on the total prevalence of all common relation types, as shown in the following equation:

$$RelSim(w_1, w_2) = \frac{1}{\sum\limits_{r_t} P(r_t)} \times \sum_{r_t} P(r_t) \times RelTypeSim(r_{t_{w_1}}, r_{t_{w_2}}), \tag{15}$$

where $r_t \in RTSet(w_1) \cap RTSet(w_2)$.

3.4.2 Relatedness

Relatedness can be expressed by the direct connection(s) between two terms through non-taxonomic relations. It is a reflection of the contextual bond between terms. As described in the gold standard datasets, MC [139], RG [140], WordSim [141], and MTurk [142], concepts were evaluated based on their similarity and relatedness. For example, in [141], one instruction was *"When estimating similarity of antonyms, consider them 'similar' (i.e., belonging to the same domain or representing features of the same concept), rather than 'dissimilar'"*. Therefore, it is essential to incorporate relatedness into a similarity measure, as shown in Eq. (6). A non-taxonomic path associates two terms with a relationship that is reflected by the intermediate relations between them. Intuitively, the longer the path between terms, the less related they are. Furthermore, each relation has a weight that reflects its importance to the domain. Hence, relations with higher weights indicate a stronger contextual relationship between the associated terms. Inversely, the weaker the weight between terms, the less related they are. Finally, terms that have multiple paths indicate stronger bond between them. For instance, the lexical terms (king and queen) share more than one path between them, as they are considered antonyms in addition to their respective synsets, which are member-meronyms of the synset 'royal family'. Based on our results, these terms would have a relatedness of 92% over both paths. Based on these principles, we propose a new relatedness measure based on weighted non-taxonomic relational paths.

Strategy 4

Taking the above considerations into account, we propose a relatedness measure as a function of the number of paths, their length and strength. The length of a path is a function of the number of non-taxonomic relations between terms. Strength is measured by the proposed prevalence of the path relations as defined in Section 3.3.2. As the path length increases, the relatedness decreases. On the other hand, as the accumulated weight of all relations in the path increases, relatedness increases. Therefore, relatedness needs to be a monotonically decreasing function with respect to path length, while monotonically

increasing with respect to the relations' weight. To satisfy these constraints, we propose the following relatedness measure:

$$Relatedness(w_1, w_2) = \frac{1}{n} \times \sum_{path_i=1}^{n} 1 - Distance_i(w_1, w_2), \tag{16}$$

where n is the number of paths, and the distance is calculated as shown below:

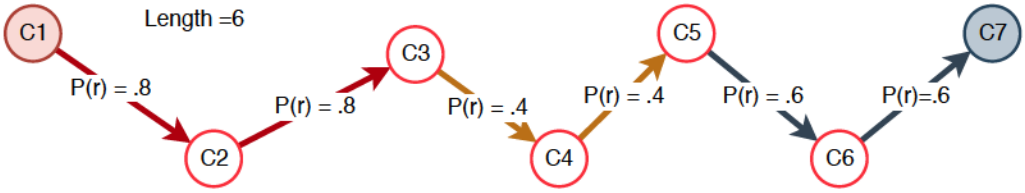
$$Distance_i(w_1, w_2) = \frac{1}{max\_depth_{wn}} \times \sum_{r_t \in path_i(w_1, w_2)} e^{-P(r_t)}, \tag{17}$$

where max_depth_{wn} is the the maximum hierarchical depth of WordNet. To ensure paths convey meaningful relatedness between terms, we only consider paths with a length shorter or equal to max_depth_{wn}. Fig. 3.7 illustrates the effects of path-length and relation-prevalence on the relatedness measure. As can be seen from Fig. 3.7a and Fig. 3.7b, where the pathways between concepts C1 and C7 have the same length but different prevelances, the higher the prevalences between concepts the greater the relatedness. However, for Fig. 3.7c, although the pathway between C1 and C7 has the same average prevalence per relation type as that in Fig. 3.7a, yet its relatedness is higher due to the shorter path. Finally, the overall relatedness measure between concepts C1 and C7 over all of the three paths is 0.854, which represents the average relatedness over all of the paths as shown in Eq. 16.

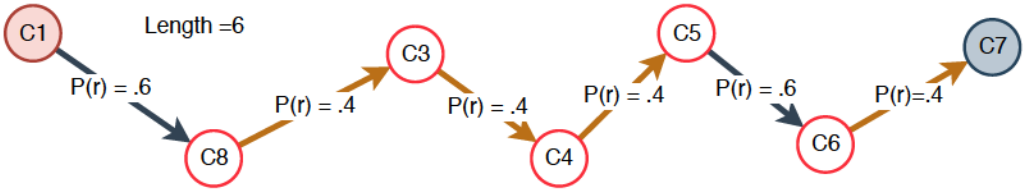
3.5 Evaluation and Experimental Results

3.5.1 Experimental Setup

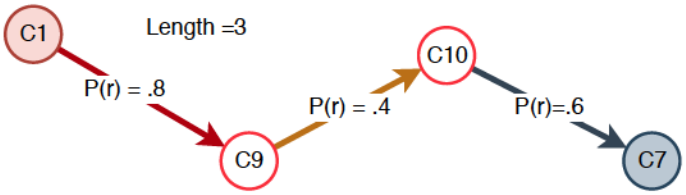
The goal of our experiments is to evaluate the proposed semantic similarity and relatedness based on the WordNet database and the most commonly-used gold standard datasets. This section presents WordNet KG, datasets, evaluation metrics, implementation, and detailed discussion of the results.



(a) 1-Distance(C1,C7) = 0.833



(b) 1-Distance(C1,C7) = 0.811



(c) 1-Distance(C1,C7) = 0.916

Figure 3.7: Relatedness paths between concepts with different relations' prevalence

3.5.1.1 Lexical Knowledge Base and Datasets

WordNet WordNet is an English-based lexical database, where words are organized within each POS category into sets of cognitive synonyms named synset. Synsets are organized into a hierarchical structure (taxonomy) from the most abstract concepts (i.e., entity in the nouns category) to the most specific leaf concepts. Synsets and lexical terms are linked through means of pointers that represent a specific semantic relationship between them.

The main and most structured explicit relation type is the hyponymy (ISA) relation and its inverse hypernymy relation, which forms the hierarchical structure of WordNet. ISA represents the generalization/specialization relation between concepts. The next most common relation used is holonomy/meronymy (part of) relation. All other relations such as antonymy, theme and derivation are used much less frequently. Tables 2.3 and 3.1

Table 3.1: Hierarchical relations in WordNet

Relation Name	**Frequency**	**Prevalence**
hyponym	75180	44.89 %
hypernym	75139	44.86 %
instance_hyponym	8592	5.13 %
instance_hypernym	8568	5.12 %

depict the frequency of each relation used within WordNet[2].

WordNet was designed with the intention of providing a machine-readable dictionary that determines a word definition through semantic relations [42]. In addition to the explicit relations, lexical terms within one synset also share implicit relations between each other, namely "synonymy". As shown in Table 2.3, this relation is explicitly emphasized in our approach. We use the synset_member relation that exists in WordNet to identify the synonyms of each synset. The number of synonym relations is then computed using the combination $\binom{n}{2}$ where n is the number of synonyms (synset_member) within one synset. Hence, the use of synonymy relation comply with the spirit of WordNet, providing semantic definitions for its synsets and lexical terms. Therefore, in this work, we employ both of explicit and implicit semantic relations to obtain better semantic similarity and relatedness between terms.

Gold Standard Datasets Our experimental environment consists of the lexical database WordNet 3.1 as a KG and three widely-used evaluation datasets.

- RG [140]: contains 65 pairs of words, where each pair is assigned a rating of similarity ranging between [0,4]; 0 being most dissimilar/unrelated terms, and 4 being exactly similar.
- MC [139]: contains 28 pairs of words, chosen carefully from RG to represent a full range of similarity/relatedness.

[2]The frequency is calculated based on the nouns POS from WordNet v3.1

- WordSim [141]: contains 353 pairs of terms. The rating range is between [0,10], where 0 is completely dissimilar/unrelated and 10 is exactly similar. The instructions provided to the subjects — humans experts — were to assign a rating for each pair based on the similarity/relatedness of the two words. More specifically, in WordSim, the instructions were clear for the antonym words to be considered as similar rather than dissimilar, since they belonged to the same domain and/or express features of the same topic.

- MTurk [142]: intended to emphasize relatedness between terms. MTurk contains 771 pairs or words with their average relatedness score between [1-5], where 1 represents 'not related' words, and 5 represents 'highly related' words

Table 3.2 provides a summary of the main characteristics and statistics of each dataset. In our implementation, we examine all pairs within each dataset, with the exception of WordSim as it contains few pairs that were not found in WordNet-3.1 due to variation in the term morphology or tense. The last four rows in the table (rows 5, 6, 7, and 8) are the focus of our poly-relational approach. Row 5 shows the number of pairs that share one or more non-taxonomic relation type, regardless of whether or not the object of the relation is the same. For instance, "car has *part-meronym* ..." and "auto has *part-meronym*...", hence both terms share the same relation type *part-meronym*. Row 6 counts the number of terms where one or more of its non-taxonomic relations contain multiple instances. For example, for "car has *part-meronym* fender, car has *part-meronym* engine", there are two instances of the relation *part-meronym*. Row 7 shows the number of pairs where at least one of its terms matches a term contained in row 6. Finally, row 8 describes the number of pairs that have at least one non-taxonomic path that connects the pair (i.e., "Car is *synonym* of auto", "King is member-meronym of royal family, and royal family is member-holonym of queen".

Table 3.2: Gold standard datasets characteristics

	Criteria	MC	RG	WordSim	MTurk
1	**# of pairs**	28	65	353	771
2	**# of pairs (our implementation)**	28	65	342	770
3	**Distinct terms/senses**	44	65	500	1281
4	**Terms with non-taxo. relations**	18	23	207	487
5	**Pairs with common relations (CR)**	4	4	54	110
6	**Terms with multi-instances (MI)**	8	10	77	204
7	**Pairs with CR & MI**	2	2	20	48
8	**Pairs with one or more path(s)**	6	12	35	149

3.5.2 Evaluation Metrics

The practice of evaluating semantic similarity measures has relied on the correlation between the proposed method and gold standard [9, 32, 33, 34, 43, 64, 65]. Two main correlation methods have been applied. The first is the Pearson correlation for two random variable X and Y, as shown below:

$$\rho = \frac{\text{cov}(X, Y)}{\sigma_x \sigma_y} \tag{18}$$

Yet, a simplified estimated correlation of the Pearson has been normally applied based on the following:

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}, \tag{19}$$

where n is the size of the sampled sets, x_i, y_i are the i^{th} element of semantic similarities reported by any measure, and the human judgment, respectively, and $\overline{x}, \overline{y}$, represent the mean for each set.

The second correlation method is the Spearman correlation coefficient (Spearman's

rho). This is a rank-based correlation, which is not restricted to continuous data.

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}, \tag{20}$$

where d is the pairwise distance of the ranks of the elements x_i and y_i, and n is the size of the sampled set.

3.5.3 Implementation

Our experimental implementation utilizes the .NET framework and dotNetRDF library[3]. This library helps to convert the N-Triples of WordNet 3.1[4] KG file into an in-memory RDF graph. Out of the 5.5 million triples, we focus only on noun synsets and lexical senses, including their relations. However, some of these relations, such as translation, gloss, and tag counts, are also ignored because they serve other purposes that are outside the scope of this research. In summary, we focus on triples with relations that serve the semantic of the noun synsets and lexical senses. These relations are listed in Tables 2.3 and 3.1. After the extraction and cleansing, we are left with a total of 81,816 noun synset, 262,786 individual lexical senses, and 374,453 instances relations. These relations are grouped into 12 non-taxonomic and four taxonomic relation types as shown in Tables 2.3 and 3.1, respectively.

It is worth mentioning that the ontology of WordNet[5] is very abstract, and the relations have very shallow hierarchy with no specific domain and range concepts. Hence, in our experiment, the taxonomic IC ($IC_{Tax}(r_t)$), and the global IC ($IC_{GC}(r_t)$) do not contribute towards calculating the full RIC ($RIC(r_t)$). As a result, relations attain their information strictly from the local IC ($IC_{LC}(r_t)$) from WordNet's KG. Based on this, The RIC in Eq. 4 will be fully equal to the local IC ($IC_{LC}(r_t)$) and the values of $\alpha = \beta = 0$, while $\gamma = 1$.

The rest of the implementation consists of an offline module that calculates WordNet's KG statistics, and the gold standard datasets statistics. In addition, we implemented the existing baselines' IC and taxonomic similarity measures.

[3] `http://www.dotnetrdf.org/api/html/N_VDS_RDF.htm`

[4] `http://wordnet-rdf.princeton.edu/`

[5] `http://wordnet-rdf.princeton.edu/ontology`

3.5.4 Experimental Results and Performance Analysis

The main purpose of this experiment is to confirm the observations in section 1.2, which emphasize that all relations convey an important informative component of a concept's semantic definition and information content, and that relation type prevalence reflects its relevance to the modeled domain. This is demonstrated in this section by showing the enhancements provided by our poly-relational approach to existing taxonomic-based similarity measures. The common practice of evaluating any approach is to compute the pairwise similarities of a given dataset. Then, calculate the correlation as discussed in section 3.5.2, with the gold standard for that dataset. The higher the correlation with the gold standard, the better the approach. Therefore, for each baseline, we implement its IC measure to evaluate non-taxonomic relations. Then employ Eq. (22) to incorporate the proposed $RelSim(w_1, w_2)$. Finally, compare our results with each baseline based on gold standard correlations.

In section 3.4 we presented four strategies, which will be referred to in this Section as S1, S2, S3, and S4. The results of each strategy, as shown in Fig. 3.8 to Fig. 3.12, are described by the gain between the proposed approach and each baseline. The gain is based on the correlation of each approach with the gold standard, which is computed as follows:

$$gain = \frac{Corr()_{PolyR} - Corr()_{baseline}}{Corr()_{baseline}}, \tag{21}$$

where $Corr()_{PolyR}$ and $Corr()_{baseline}$ are the correlations of the PR-SSR approach and any given baseline approach with the gold standard dataset, respectively. Each figure shows the gain for its respective strategy, and includes the evaluation using the Pearson correlation, as defined in Eq. (19), and the Spearman correlation, as defined in Eq. (20).

Finally, we compute the full semantic similarity and relatedness as described in Eq. (6). However, since S1, S2, and S3 do not involve relatedness, semantic similarity (*SemSim*) for these strategies includes only the taxonomic similarity and the non-taxonomic relational similarity measures, as shown in the equation below:

$$SemSim(w_1, w_2) = (1 - \alpha_1) \times TaxSim(w_1, w_2) + \alpha_1 \times RelSim(w_1, w_2), \tag{22}$$

where α_1 is the contribution of the relational similarity that overcomes the limitations of the existing baseline taxonomic similarity. This is measured by the prevalence of the non-taxonomic relations common to both terms, as in the following equation:

$$\alpha_1 = \sum_{r_t \in R(w_1) \cap R(w_2)} P(r_t) \tag{23}$$

Strategy 4, on the other hand, incorporates the relatedness in the overall semantic similarity and relatedness measure.

Strategy 1

As stated in Section 3.4.1, S1 considers all non-taxonomic relations and encapsulates their SemIC values into a single attribute for each term, namely SemIC. The results for this strategy, as presented in Fig. 3.8, demonstrates limited improvements across the evaluated baselines and gold standard datasets. For instance, using MC gold standard, although our approach demonstrates a gain of 0.05% with *Meng* as a baseline, it shows a consistent decline with the two other datasets, especially when using the Spearman correlation. Similarly, with MC gold standard using *Seco* as a baseline, the gain is approximately 0.06%, while it shows a decline of 0.27% with the Spearman correlation. Examining the same baseline with WordSim gold standard dataset, our approach has approximately 0.64% and 1.10% gain with both the Pearson and the Spearman correlations, respectively.

The decline in this strategy is due to the fact that existing taxonomic IC-based similarity measures rely on the IC of the LCS of the two terms, as shown in in Table 2.2. However, non-taxonomic relations do not follow hierarchical structure, and therefore, traditional taxonomic IC-based similarity measures are not effective for evaluating SemIC. This was the main motivation for investigating new ways for computing relational-based similarity in subsequent strategies. Furthermore, S1 does not take into consideration the type of non-taxonomic relations when comparing the semantic IC values of two terms. Thus, ignoring an important aspect of terms' semantic.

Strategy 2

Strategy 1 has clearly undermined baselines similarities instead of improving them. As discussed in Section 3.4.1, S2 overcomes the limitations of S1 by expanding the encapsulated SemIC of a term. Instead of reducing SemIC to a single-value attribute, terms are attributed a vector of relation type-based RIC measures, each of which represents a single relation type (part-meronym, instance-meronymy, derivation, etc.), as shown in Eq. (9).

As shown in Fig. 3.9a, with the exception of MTurk dataset using *Seco, Zhou, Meng*, and *sánchez*, S2 demonstrates a consistent gain across all baselines and gold standard datasets. For instance, using WordSim gold standard dataset and *Meng* as a baseline, S1 shows a decline of -0.58%, while S2 shows a gain of 0.50%. This remains consistent with the Pearson correlation using the other datasets too. However, based on the Spearman correlation, as shown in Fig. 3.9b, although there is an overall improvement with RG and WordSim gold standard datasets, MC still demonstrates a decline with most baselines. We believe this is due to the sensitivity of the Spearman ranking correlation on such a small dataset, as the same baselines, with a relatively larger dataset (RG), still demonstrate a minimal gain.

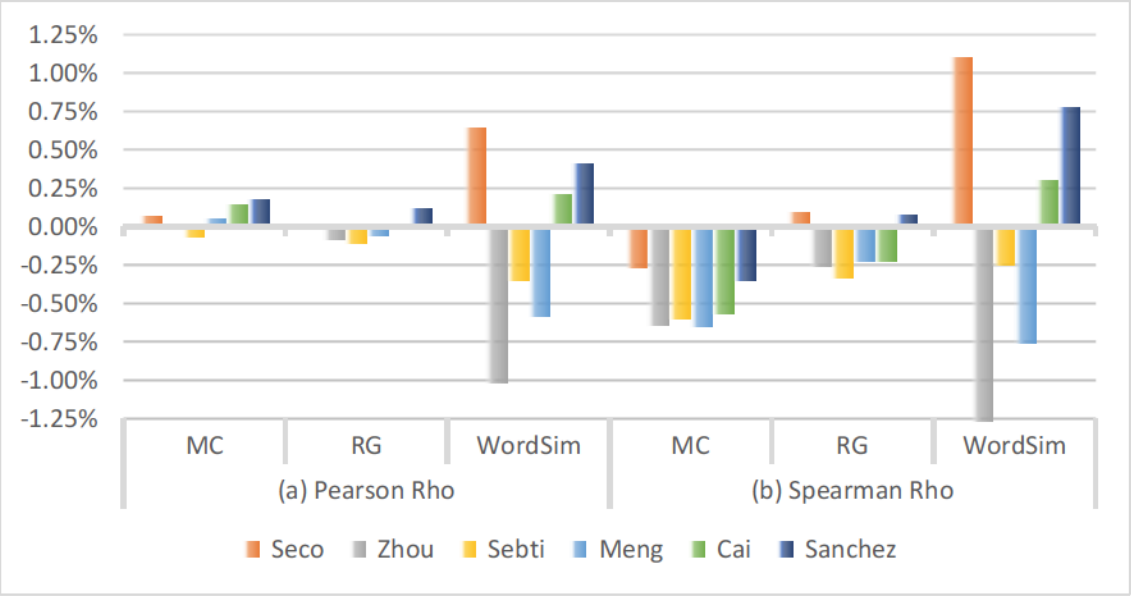


Figure 3.8: Semantic similarity gain using strategy 1

Despite the decline in MC dataset using the Spearman correlation, improvements in all other evaluations motivated us to go beyond and expand each vector element of SemIC into existing instance relations of that type. This is the main motivation behind S3.

Strategy 3

Fig. 3.10a shows the gain of S3 using the Pearson correlation. It can be observed that the gain nearly mirrors S2 results. This can be explained by the fact that only a small proportion of the datasets used exhibit pairs with common relations that have multiple instances, as shown in Table 3.2. Therefore, their effect is not significant on the overall results. On the other hand, using the Spearman correlation, a significant improvement can be observed for MC and RG datasets as shown in Fig. 3.10b. This can be justified by the fact that the Spearman correlation, being based on ranking, is very sensitive to the size of the dataset used. For a small dataset, semantic similarity changes to a few pairs will significantly impact the overall ranking, thus resulting in considerable improvement to the correlation. Inversely, for a large dataset, semantic similarity changes to a few pairs will not have the same impact on the overall ranking, thus resulting in minor improvement

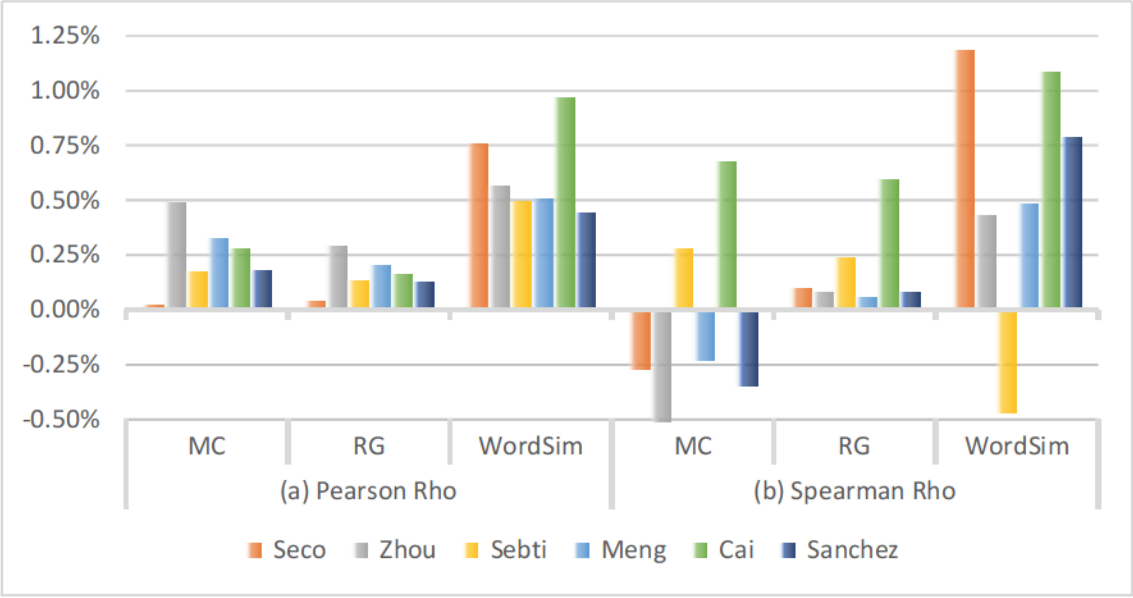


Figure 3.9: Semantic similarity gain using strategy 2

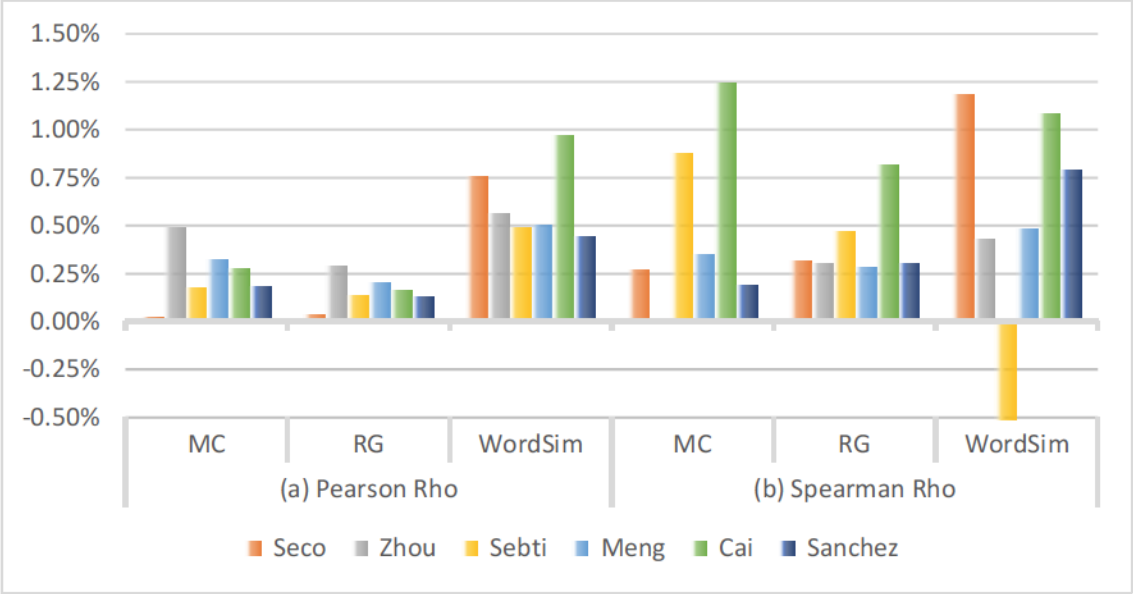


Figure 3.10: Semantic similarity gain using strategy 3

of the correlation. This is clearly shown in Fig. 3.10b for MC and RG, which are small datasets with only 2 pairs affected by semantic similarity changes. However, for WordSim dataset, which includes 342 pairs, and only 20 pairs affected by semantic similarity change, the improvement in correlation is very minor.

The only surprising mystery remaining is a decline using the *Sebti* baseline with WordSim dataset. After analyzing the baseline, dataset, and other literature [58], we conclude that this could be caused by a small subset of the dataset that is affected by *Sebti's* calculation method. This finding is confirmed in another literature [58], noticing similar behaviour of abnormality with the same dataset. To address the shortcomings in WordSim dataset, we conducted another experiment using only a subset of the whole dataset, focusing mainly on relevant pairs. These are pairs with at least one common relation type, or at least one connected path, or both. In particular, 54 pairs were used to test strategies S1, S2, and S3, and 66 pairs were used to test S4. The use of this semantically rich dataset shows consistent gain across S2, S3, and S4 as can be observed in Fig. 3.11. On the other hand, the inconsistent gain in S1 confirms our initial observation for S1, that semantic-blind comparison is not effective.

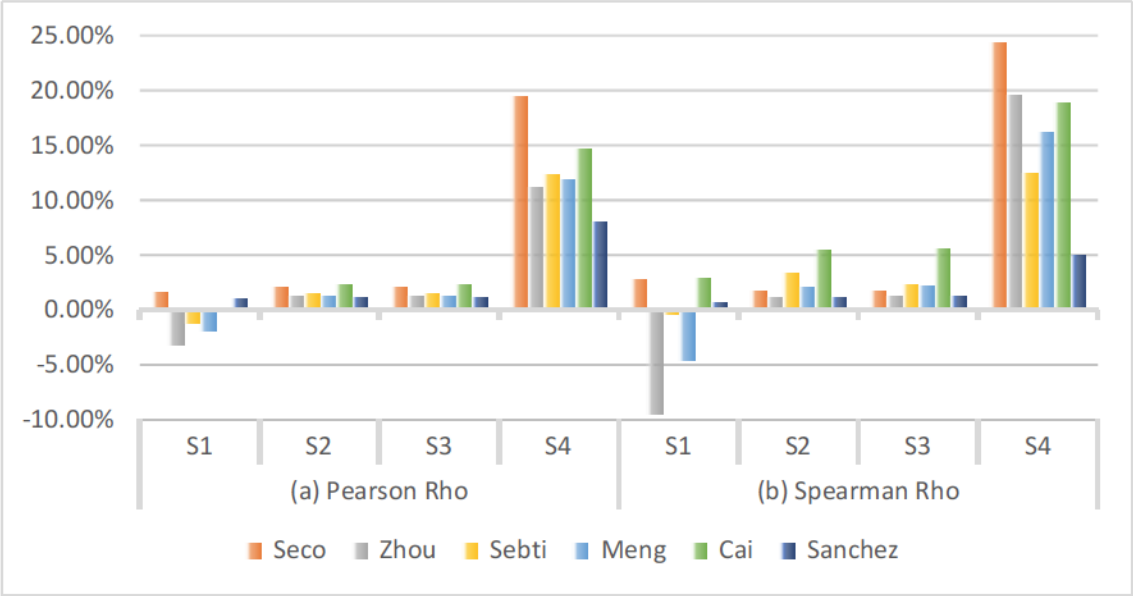


Figure 3.11: Semantic similarity gain for all strategies using WordSim relevant pairs

Strategy 4

As described in Section 3.4.1, S4 differs from the previous strategies by incorporating a relatedness measure, which is based on weighted paths between terms, as defined in Eq. (16). The overall semantic similarity and relatedness measure that was defined in Eq. (6),

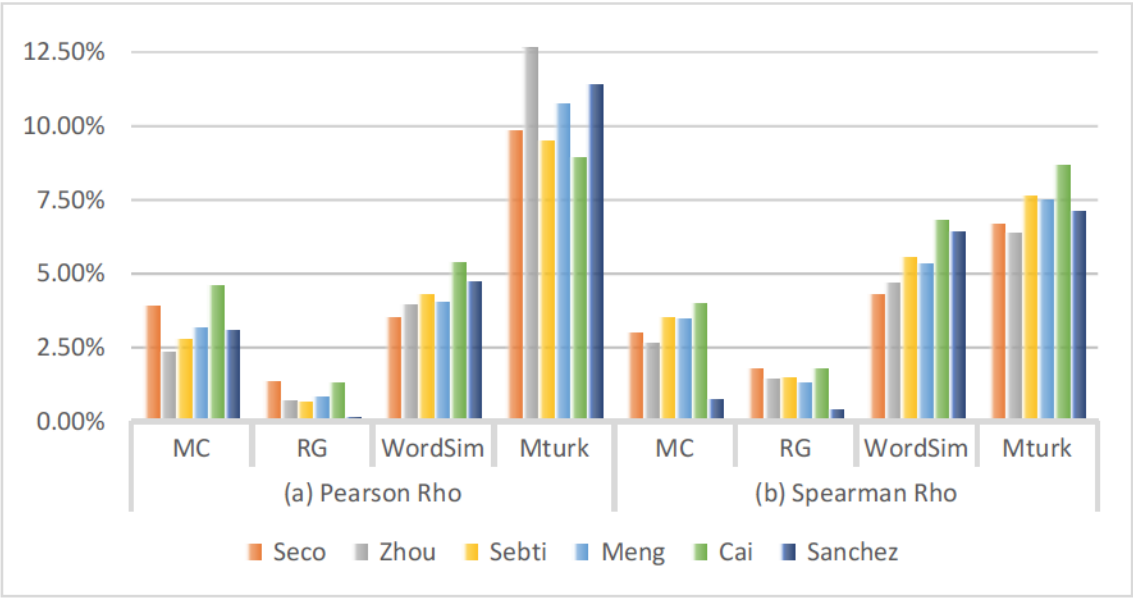


Figure 3.12: Semantic similarity and relatedness gain using strategy 4

is expressed as follows:

$$SemSimRel(w_1, w_2) = (1 - \alpha_2 - \beta) \times TaxSim(w_1, w_2) + \alpha_2 \times RelSim(w_1, w_2) + \beta \times Relatedness(w_1, w_2), \quad (24)$$

where α_2 and β measure the contribution of relational similarity and relatedness, respectively. Note that α_2 in Eq. (24) is different from α_1 in Eq. (22), due to the effect of the relatedness parameter in S4. The optimal values for α_2 and β in S4 are empirically evaluated. We used different combinations of (α_2, β), and we found that the optimal values across all datasets and baselines are $\alpha_2 = 0.12$ and $\beta = 0.55$.

To test strategy 4 with a more relevant dataset that includes both similarity and relatedness, we used the MTurk gold standard dataset. As described in section 3.5.1.1, MTurk was built to capture relatedness measure between terms, which is the main focus of strategy 4. Based on the results shown in Fig. 3.12, the gains for this strategy are consistent across all baselines and gold standard datasets. For example, using *Sebti* as a baseline, S3 shows a decline with the Spearman correlation, while S4 shows over 5% of gain. Also, the results show better performance with MTurk, which is a larger relatedness dataset, thus confirming scalability of S4. As shown in Fig. 3.12, the highest gain is 12.63%, which is achieved based on *Zhou* baseline with MTurk gold standard dataset using Pearson's correlation. On the other hand, the highest correlation value for the same dataset is attained using *Cai* baseline with the pearson correlation of 0.8620, see Table 3.3. As shown in Fig. 3.12, the proposed poly-relational approach demonstrates consistent improvement in the semantic similarity and relatedness measure across all datasets. These results are coherent with the human perception of semantic similarity and relatedness within the gold standard datasets.

To test the robustness of the proposed approach, we further extended our experiment to include WordNet's existing similarity measures [143], which includes some corpus-based measures, such as RES [52], WUP [60], and LIN [63], as well as taxonomic-based path measure PATH [59]. Furthermore, we compared our approach with the state of the art Knowledge Graph Embedding semantic similarity models implemented in [71]. We used the implementation provided in [71] to train three models (TransE, TransH, and TransG)

Table 3.3: Pearson correlation with gold standard and proposed strategies

Method	**MC**	**RG**	**WordSim**	**MTurk**
RES_{corpus}	0.8154	0.8475	0.5370	0.7240
WUP_{corpus}	0.7740	0.8047	0.4256	0.6735
LIN_{corpus}	0.7759	0.7569	0.4376	0.6151
$PATH_{\text{is-a}}$	0.7504	0.7889	0.4295	0.6732
$Seco_{\text{graph}}$	0.8943	0.8680	0.3528	0.4534
$Zhou_{\text{graph}}$	0.8429	0.8403	0.3259	0.4414
$Sebti_{\text{graph}}$	0.8434	0.8577	0.3835	0.4564
$Meng_{\text{graph}}$	0.8563	0.8711	0.3652	0.4702
Cai_{graph}	0.8635	0.8840	0.3856	0.4898
$sánchez_{\text{graph}}$	0.8950	0.8701	0.3475	0.4489
$TransE_{\text{corpus}}$	0.8310	0.7737	0.4032	0.4465
$TransH_{\text{corpus}}$	0.7800	0.8041	0.4097	0.4289
$TransG_{\text{corpus}}$	0.8102	0.7720	0.3639	0.4083
Strategy 1_{graph} (baseline)	0.8966 (sánchez)	0.8841 (Cai)	0.3864 (Cai)	0.4902 (Cai)
Strategy 2_{graph} (baseline)	0.8966 (sánchez)	0.8854 (Cai)	0.3894 (Cai)	0.4900 (Cai)
Strategy 3_{graph} (baseline)	0.8966 (sánchez)	0.8854 (Cai)	0.3894 (Cai)	0.4900 (Cai)
Strategy 4_{graph} (baseline)	**0.9291** (Seco)	**0.8953** (Cai)	**0.7079** (Seco)	**0.8620** (Cai)

on WordNet and the gold standard datasets. We have then obtained the embedding vectors for each term and computed the cosine similarity for each pair of terms. Tables 3.3 and 3.4 display the actual Pearson and Spearman correlations based on our implementation using WordNet 3.0, the KGE from [71] and the Natural Language Toolkit (NLTK) [144] WordNet similarity implementations. Also, the tables present the respective correlations for the six examined benchmarks [9, 32, 33, 34, 43, 65], in addition to the poly-relational approach, showing the best obtained correlation across all baselines for each strategy. It can be seen from the results provided in Tables 3.3 and 3.4 that the proposed strategies outperform all baselines. Furthermore, the results show gradual improvement from S1

Table 3.4: Spearman correlation with gold standard and proposed strategies

Calculation Method	**MC**	**RG**	**WordSim**	**MTurk**
RES_{corpus}	0.8028	0.8067	0.6082	0.7020
WUP_{corpus}	0.7681	0.7766	0.4278	0.7224
LIN_{corpus}	0.7457	0.6952	0.4751	0.6758
$PATH_{\text{is-a}}$	0.7506	0.8002	0.3869	0.7271
$Seco_{\text{graph}}$	0.8660	0.7963	0.3308	0.4896
$Zhou_{\text{graph}}$	0.8130	0.7883	0.3232	0.5057
$Sebti_{\text{graph}}$	0.7781	0.7693	0.3349	0.4762
$Meng_{\text{graph}}$	0.8001	0.7918	0.3205	0.5082
Cai_{graph}	0.8174	0.7981	0.3150	0.5025
$sánchez_{\text{graph}}$	0.8772	0.7994	0.3195	0.5036
$Zhang_{\text{graph}}$	0.3480	0.3434	0.1013	0.2195
$TransE_{\text{corpus}}$	0.8085	0.7236	0.3525	0.4622
$TransH_{\text{corpus}}$	0.7670	0.7406	0.3728	0.4454
$TransG_{\text{corpus}}$	0.8342	0.6689	0.3180	0.4203
Strategy 1$_{\text{graph}}$ (baseline)	0.8741 (sánchez)	0.8001 (sánchez)	0.3344 (Seco)	0.5035 (sánchez)
Strategy 2$_{\text{graph}}$ (baseline)	0.8741 (sánchez)	0.8028 (Cai)	0.3347 (Seco)	0.5029 (sánchez)
Strategy 3$_{\text{graph}}$ (baseline)	0.8788 (sánchez)	0.8046 (Cai)	0.3347 (Cai)	0.5039 (sánchez)
Strategy 4$_{\text{graph}}$ (baseline)	**0.8917** (Seco)	**0.8121** (Cai)	**0.6848** (Seco)	**0.8600** (Cai)

to S4, with the exception of S3 when using the Pearson correlation. This is justified above, while discussing the results of S3. As highlighted in both tables, the proposed method provides significant improvement over all baselines with both the Pearson and the Spearman correlations, thus showing its superiority over all other methods.

The complexity of the proposed semantic similarity and relatedness algorithm is dependent on the IC function for the used baseline and concepts' depth in WordNet graph. It should be noted here that the number of relation types in the WordNet graph is constant, and therefore it does not affect the computational complexity of the proposed technique. As shown in Eq. 6, the complexity of the proposed method is given by the maximum complexity among the three algorithms used for computing taxonomic similarity, relational-

based similarity and relatedness. The taxonomic similarity and relational-based similarity have the same algorithm complexity since they are both dependent on the computational complexity of the concept's IC value. Although the IC function varies from one baseline to another, yet their complexity is linear, that is $\mathcal{O}(n)$, where n is the number of hierarchical features used to compute the IC value for each baseline (e.g. hyponyms, siblings, direct hyponyms, depth of concepts in the graph) as shown in Table 2.1. However, the complexity of the relatedness measure is $O(nm)$, where n and m represent the number of paths and the maximum path length between two terms respectively. Therefore, the overall complexity of the proposed algorithm is given by $O(mn)$, which is considered a reasonable polynomial complexity compared to the state of the art KGE semantic similarity methods, where models need to be trained on large datasets using computationally expensive neural network configurations.

3.6 Conclusion

In this chapter, we examined the concept of semantic similarity based on the information content of terms. We introduced a novel approach that can be applied to any knowledge domain. The proposed approach exploits both taxonomic and non-taxonomic relations to compute IC and SemIC of all terms. These are employed at different granularity levels to measure semantic similarity. Furthermore, we introduced a new approach to measure relatedness based on weighted paths built out of non-taxonomic relations.

The experimental results prove that non-taxonomic relations add valuable information to their associated terms, and contribute to determining the semantic similarity between them. Furthermore, it was shown that prevalence of each relation type is an important ingredient in measuring semantic similarity and relatedness, thus mimicking human perception. Therefore, we can conclude that non-taxonomic relations play a vital role in determining domain specific semantic similarity.

Chapter 4

Semantic Word Sense Disambiguation

4.1 Introduction

Regardless of the language used, spoken words carry a specific meaning that allows humans to communicate and understand each other. Communication would be confusing if the words used in a sentence carry multiple meanings, which could be clarified by providing additional context.

This chapter presents the main tasks of pursuing the WSD challenge (Section 4.2). Section 4.3 presents in details the proposed WSD process including the novel Sequential Contextual Similarity Matrix Multiplication (SCSMM) and back-tracing algorithms. This section also includes a detailed flowchart of the proposed system. Section 4.4 presents a description of the leading gold standard datasets and a thorough evaluation of the proposed approach.

4.2 Word Sense Disambiguation Tasks

The research community and the SemEval International Workshop on Semantic Evaluation[1] described two main tasks for WSD. These tasks are: Lexical Sample (LS) and AW.

[1]Current workshop website: `http://alt.qcri.org/semeval2020/`

The main difference between the two is the number of target terms to be disambiguated within a sentence, more details as follows:

Lexical Sample (LS) : This is mostly applied by supervised approaches, where the system is required to disambiguate a small set of predefined ambiguous words, typically one target word per sentence, with the rest of the words providing a context. Having only one target word per sentence simplifies the supervised classification task; hence such approaches, in general, achieve higher accuracy, as they are focused on a single class with a broader context.

All-Words (AW) : This task assumes open-class words and a smaller context. The objective is to disambiguate all open words within a sentence, leaving a smaller concrete context. Supervised systems may suffer while solving this task due to a lack of a sense-annotated training dataset covering a full lexicon. On the other hand, other approaches, specifically knowledge/dictionary-based approaches, are more suitable for this task since they can exploit a full lexicon knowledge base on demand with very little or no prior training [37].

Although AW is considered a more realistic task to evaluate, yet, producing a training corpus for a LS task is much easier; since human annotators have to read the senses' definitions once for a block of instances for the same target word. On the other hand, to produce a training corpus for AW task, human annotators have to read the definitions for each word in the sequence with every annotated sense. Nonetheless, very few datasets have been sense-annotated for AW WSD task, and only one is manual. Besides, more gold-standard evaluation datasets have been presented through the SensEval/SemEval international workshop from 1998 to date. Section 4.4.1.1 presents these datasets in more detail.

4.3 Proposed Method

This section presents a novel, context-aware WSD algorithm based on a KG semantic similarity and relatedness measure. Our main intuition is derived from the brain's basic steps to analyze and disambiguate words in context (i.e., sentence, document).

As described in Section 2.3.4, four main elements are essential for disambiguating words within a sentence. These are (i) the sequence of the terms within the sentence; (ii) the connectivity between various concepts (i.e., senses) of ambiguous terms; (iii) a basic heuristic knowledge of each term and its various concepts (i.e., senses); and (iv) the broader context of the document. These design elements are incorporated into the proposed WSD algorithm.

4.3.1 System Flowchart

Fig. 4.1 describes the main tasks of the proposed WSD method, starting from parsing the XML content of the dataset and the NLP preprocessing tasks, followed by the construction of document's context. The document context consists of all context words within each document (terms with a single sense) that have nonzero TF-IDF value. Then, the three main WSD processes, which make up the WSD algorithm, are executed for each sentence in the document. These are the construction of Contextual Similarity Matrix (CSM)s queue, followed by the main SCSMM algorithm, and finally the identification of the most contributing senses to the global context in the back-tracing algorithm. In the cases where there is any ambiguous terms left, the carry-forward process is executed to disambiguate them.

4.3.2 WSD Algorithm

The complete WSD process, as described in Algorithm 2, consists of the CSM queue construction, a novel SCSMM and a back-tracing algorithms for an AW WSD task. The proposed method follows a knowledge-based approach using WordNet as a sense dic-

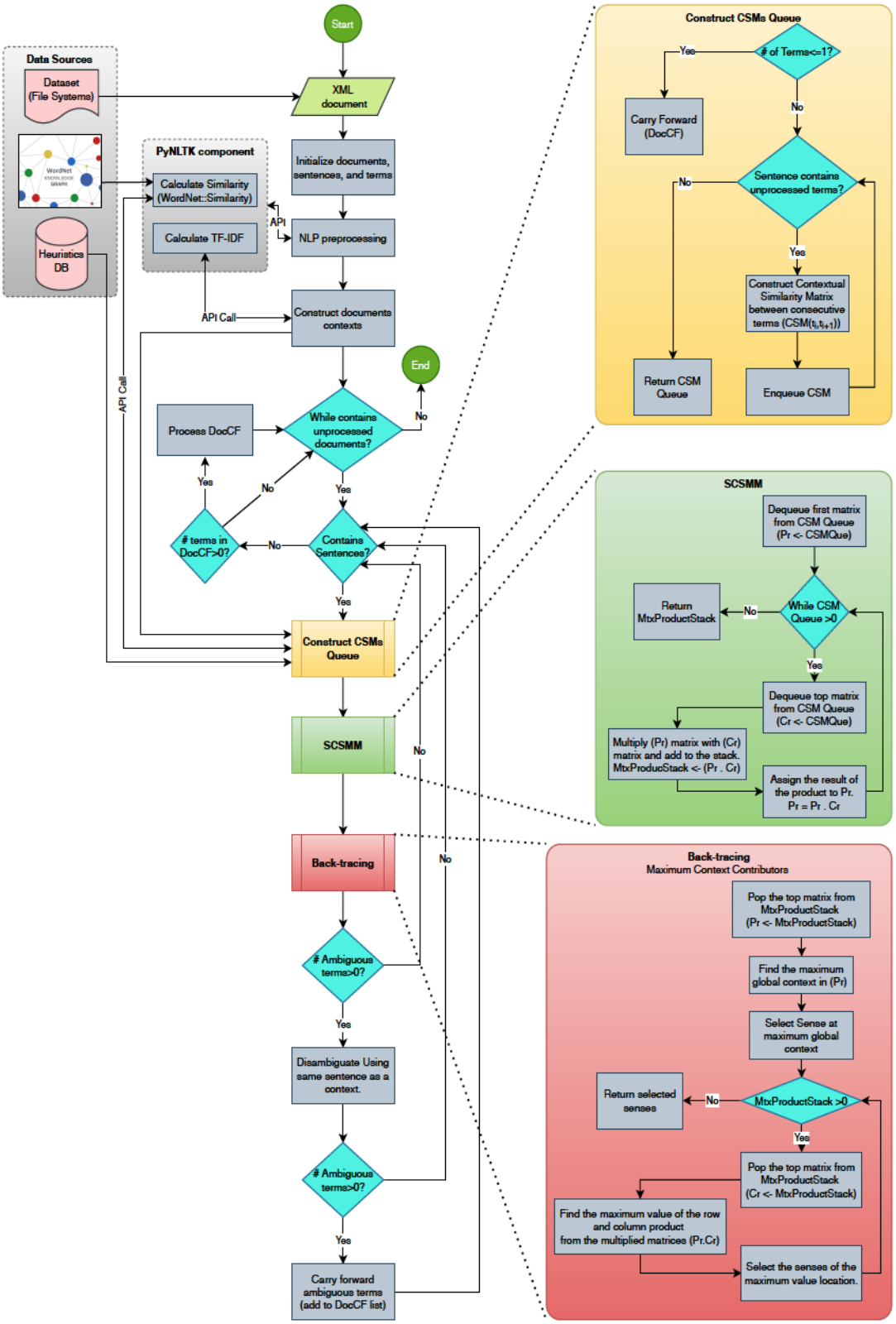


Figure 4.1: Flowchart for the proposed WSD algorithm

tionary and the main knowledge resource. Before starting the WSD process, standard NLP preprocessing steps take place, such as sentence tokenization, stop-words removal, lemmatization, and POS tagging.

Algorithm 2: WSD Algorithm Using SCSMM

Input : S: Sentence with list of ambiguous words

Output: $\hat{S}$: Sentence with annotated sense

1 **Data Structures**:

2 $CSMQue$: Contextual Similarity Matrices Queue

3 $MtxProductStack$: A Stack for the produced matrices resulting from the product of consecutive matrices

4 **for** $i \leftarrow 0$ *to* $(|TermsOf(S)| - 1)$ **do**

5 $\quad CSMQue \xleftarrow{Enqueue}$ Call $getSemSimMatrix(S_i, S_{i+1})$

6 $MtxProductStack \leftarrow$ Call $SCSMM(CSMQue)$

7 $\hat{S} \leftarrow$ Call $BMCC(MtxProductStack)$

Before delving into the algorithm, the next section presents the core components that construct the CSM; these are the semantic similarity, sense heuristics, and document context.

4.3.2.1 CSM Core Components

The similarity matrix algorithm described in Algorithm 3 employs the aforementioned semantic similarity and relatedness measure as the similarity measure between the senses of every term and its consecutive term $SCM(t_i, t_{i+1})$. The local context generated by the consecutive terms' similarities is then complemented by the heuristic of each sense and the global context from the document context similarity. As a result, each cell in the CSM matrix resembles the local context, prior knowledge, and document context, see lines 7-9 in Algorithm 3.

Algorithm 3: Get Semantic Similarity matrix method

Input : Pr_{term}: First term

Cr_{term}: Second term

Output: $SimMtx$: Similarity Matrix

Data Structures:

CSM: Contextual Similarity Matrix

Initialization:

$CSM \leftarrow NewMatrix[|Sense(Pr_{term})|][|Sense(Cr_{term})|]\{0\}$

foreach $s_i \in Sense(Pr_{term})$ **do**

  foreach $s_j \in Sense(Cr_{term})$ **do**

    /* Get the Semantic Similarity and Relatedness */

    $CSM[i][j] \leftarrow SSR(s_i, s_j)$

    /* Apply heuristics as a weighted frequency of each sense */

    $CSM[i][j] *= H(s_i) * H(s_j)$

    /* Apply document context similarity of each sense */

    $CSM[i][j] *= DocCtxSim(s_i) * DocCtxSim(s_j)$

return CSM

1- Semantic Similarity: A semantic similarity and relatedness measure represents a direct and local context between consecutive terms. The main idea is to find the maximum pairwise context between senses of the two consecutive terms. However, it is possible to have more than one local context from two words based on the combination of their senses. Various knowledge-based semantic similarity and relatedness measures have been evaluated in order to determine the best similarity measure for our algorithm. These measures are presented in Section 2.2.1 and [119]. We further evaluate these measures in Section 4.4.3.

2- Sense Heuristics: In addition to the semantic similarity between senses, each sense has heuristic information that reflects its used frequency. These heuristics are observed from the available training datasets; SemCor and OMSTI. The heuristic function is based

on the senses frequency distribution within the training dataset. More formally, for a term w_i that has a set of senses $\{S\}$, and a sense $s_{ij}, 1 \leq j \leq |S|$, the heuristic function is described as below:

$$H(s_{ij}) = \begin{cases} P(s_{ij}|w_i) & , s_{ij} \in \{S\} \\ \frac{1}{Count(w_i)} & , s_{ij} \notin \{S\} \\ 1 & , w_i \notin \{W\} \end{cases}, \tag{1}$$

where $P(s_{ij}|w_i)$ is the conditional probability of the sense s_{ij} given its term w_i, that is computed based on their respective counts within the dataset as follows:

$$P(s_{ij}|w_i) = \frac{Count(s_{ij})}{Count(w_i)} \tag{2}$$

Note that if the training dataset does not contain the term w_i, its heuristic is set to one, and it will not affect the similarity matrix.

3- Document Context: As described in the semantic similarity, multiple sense-pairs might have high similarity indicating various contexts. To determine the appropriate context in the sentence, we crosscheck each sense with the document context obtained from all non-ambiguous terms in the document. Formally, for a given document with sets of ambiguous and non-ambiguous (context) terms $D = \{\{A\} \cup \{C\}\}$, and each ambiguous term w_i ($w_i \in \{A\}$) has a set of senses $\{S_{w_i}\}$, then the sense s_{ij} ($s_{ij} \in S_{w_i}$) has a context similarity weight $weight_{CtxD}(s_{ij}|C)$ with the document context C expressed as the average similarity with all context terms $c_k \in \{C\}$ as depicted in the equation below:

$$weight_{CtxD}(s_{ij}|C) = \frac{1}{|C|} \times \sum_{c_k \in C} sim_{jcn}(s_{ij}, c_k) \tag{3}$$

Illustrative Example: Consider the sentence *"I'm **walking** to the **bank**"*, with the two ambiguous words *'walk'* and *'bank'*. The similarity matrix, Table 4.1, shows high similarities between the sense pairs $walk_v^9 - bank_n^3$, and $walk_v^7 - bank_n^2$ of 0.092 and 0.077,

respectively. These represent a local context for each pair of senses. For more details of these senses and their definitions, see Fig. 4.2.

Table 4.1: Similarity matrix between terms $walk_v$ and $bank_n$

	bank1	bank2	bank3	bank4	bank5	bank6	bank7	bank8	bank9	bank10
walk1	0.051	0.053	0.047	0.045	0	0	0	0	0	0
walk2	0.048	0.044	0.044	0.037	0	0	0	0	0	0
walk3	0.069	0.072	0.063	0.059	0	0	0	0	0	0
walk4	0.042	0.039	0.039	0.033	0	0	0	0	0	0
walk5	0.069	0.072	0.063	0.059	0	0	0	0	0	0
walk6	0.065	0.068	0.060	0.056	0	0	0	0	0	0
walk7	0.067	**0.077**	0.061	0.058	0	0	0	0	0	0
walk8	0.066	0.075	0.061	0.058	0	0	0	0	0	0
walk9	0.088	0.069	**0.092**	0.055	0	0	0	0	0	0
walk10	0.065	0.063	0.059	0.053	0	0	0	0	0	0

For our system to disambiguate such a small sentence with no additional context, it relies only on the semantic similarity. Therefore, the senses $walk_v^9$ and $bank_n^3$ would be selected since they have the highest similarity of 0.092 compared to all other combinations. However, when adding heuristics, the results change completely towards another pair $walk_v^1$ and $bank_n^2$ with the highest similarity of 0.0236. Intuitively, people would think that the first meaning of walk ($walk_v^1$) and one of the first two senses of $bank$ would be more meaningful contexts than the rest. This intuition is clearly visible in Table 4.2 with the top two senses of $bank$ ($bank_n^1$ and $bank_n^2$). Note that the heuristic weights for $walk_v^1$ is 0.9, and for $bank_n^1$ and $bank_n^2$ are 0.35 and 0.5, respectively. Heuristics were computed using both of SemCor and OMSTI datasets.

Finally, if we learn additional context around the sentence, such as non-ambiguous terms within the same document (i.e., river), our brain will shift towards a more concrete context based on the document's main topic, and thus does our system. The first sense will have higher similarity than the second one, with the first sense $walk_v^1$ of 0.153 and

receives four balls) *"he worked the pitcher for a base on balls"*
- S: (n) **walk**, manner of walking (manner of walking) *"he had a funny walk"*
- S: (n) **walk** (the act of walking somewhere) *"he took a walk after lunch"*
- S: (n) **walk**, walkway, paseo (a path set aside for walking) *"after the blizzard he shoveled the front walk"*
- S: (n) **walk** (a slow gait of a horse in which two feet are always on the ground)
- S: (n) walk of life, **walk** (careers in general) *"it happens in all walks of life"*

Verb

- S: (v) **walk** (use one's feet to advance; advance by steps) *"Walk, don't run!"; "We walked instead of driving"; "She walks with a slight limp"; "The patient cannot walk yet"; "Walk over to the cabinet"*
- S: (v) **walk** (accompany or escort) *"I'll walk you to your car"*
- S: (v) **walk** (obtain a base on balls)
- S: (v) **walk** (traverse or cover by walking) *"Walk the tightrope"; "Paul walked the*

WordNet Search - 3.1

- WordNet home page - Glossary - Help

Word to search for: bank Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- S: (n) **bank** (sloping land (especially the slope beside a body of water)) *"they pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents"*
- S: (n) depository financial institution, **bank**, banking concern, banking company (a financial institution that accepts deposits and channels the money into lending activities) *"he cashed a check at the bank"; "that bank holds the mortgage on my home"*
- S: (n) **bank** (a long ridge or pile) *"a huge bank of earth"*
- S: (n) **bank** (an arrangement of similar objects in a row or in tiers) *"he operated a bank of switches"*
- S: (n) **bank** (a supply or stock held in reserve for future use (especially in emergencies))
- S: (n) **bank** (the funds held by a gambling house or the dealer in some gambling games) *"he tried to break the bank at Monte Carlo"*
- S: (n) **bank**, cant, camber (a slope in the turn of a road or track; the outside is higher than the inside in order to reduce the effects of centrifugal force)
- S: (n) savings bank, coin bank, money box, **bank** (a container (usually with a slot in the top) for keeping money at home) *"the coin bank was empty"*
- S: (n) **bank**, bank building (a building in which the business of banking transacted) *"the bank is on the corner of Nassau and Witherspoon"*
- S: (n) **bank** (a flight maneuver; aircraft tips laterally about its longitudinal axis (especially in turning)) *"the plane went into a steep bank"*

Verb

(b) Noun senses for the term bank

- S: (v) **bank** (tip laterally) *"the pilot had to bank the aircraft"*
- S: (v) **bank** (enclose with a bank) *"bank roads"*
- S: (v) **bank** (do business with a bank or keep an account at a bank) *"Where do you bank in this town?"*
- S: (v) **bank** (act as the banker in a game or in gambling)
- S: (v) **bank** (be in the banking business)
- S: (v) deposit, **bank** (put into a bank account) *"She deposits her paycheck every month"*
- S: (v) **bank** (cover with ashes so to control the rate of burning) *"bank a fire"*
- S: (v) count, bet, depend, swear, rely, **bank**, look, calculate, reckon (have faith or

Figure 4.2 Definitions for ‘walk’ and ‘bank’ in WordNet

Table 4.2: Similarity matrix with heuristics between terms $walk_v$ and $bank_n$

	bank1	bank2	bank3	bank4
walk1	0.0158	**0.0236**	0.0021	0.0010
walk2	0.0003	0.0004	0.0000	0.0000
walk3	0.0004	0.0006	0.0001	0.0000
walk4	0.0001	0.0001	0.0000	0.0000
walk5	0.0001	0.0002	0.0000	0.0000
walk6	0.0001	0.0002	0.0000	0.0000
walk7	0.0001	0.0002	0.0000	0.0000
walk8	0.0001	0.0002	0.0000	0.0000
walk9	0.0002	0.0002	0.0000	0.0000
walk10	0.0001	0.0002	0.0000	0.0000

0.151, respectively. The final correct senses in this case would be $walk_v^1$ and $bank_n^1$. On the other hand, if the document contained more financial terms (i.e., central_bank), the other sense would be selected. Based on the above, we employed the document's context similarity, which improves the overall similarity between the senses.

4.3.2.2 Sequential Contextual Similarity Matrix Multiplication Algorithm

Once all CSMs are constructed for the sentence, the WSD algorithm starts by building a similarity matrix queue ($CSMQue$) from all CSMs, maintaining their sequence, see Algorithm 2 lines 4-5. Line 6 in the algorithm generates the final matrix based on the sequential multiplication of the matrices as presented in the SCSMM algorithm (Algorithm 4). Fig. 4.3 illustrates the sequential multiplication process of the consecutive local CSMs for a sample sentence with four ambiguous words. Finally, the algorithm applies a back-tracing process to determine the most contributing senses to the maximum global context. Next, we describe in detail the SCSMM algorithm followed by the back-tracing algorithm.

Algorithm 4: Sequential Contextual Similarity Matrix Multiplication

Input : $CSMQue$: Contextual Similarity Matrices Queue

Output: $MtxProductStack$: A stack stores the product of consecutive matrices

1 **Data Structures**:

2 Pr_{matrix}: Stores the previous matrix

3 Cr_{matrix}: Stores the current matrix

4 $MtxProductStack$: A stack stores the product of consecutive matrices

5 **Initialization**:

6 $Pr_{matrix} \xleftarrow{Dequeue} CSMQue$

7 $MtxProductStack \xleftarrow{Push} Pr_{matrix}$

8 **while** $CSMQue \neq Empty$ **do**

9 $\quad Cr_{matrix} \xleftarrow{Dequeue} CSMQue$

10 $\quad MRes \leftarrow Pr_{matrix} \cdot Cr_{matrix}$

11 $\quad MtxProductStack \xleftarrow{Push} MRes$

12 $\quad Pr_{matrix} \leftarrow Cr_{matrix}$

Result: $MtxProductStack$

Similarity Matrices Multiplication: Once all CSMs are constructed between consecutive terms (see Fig. 4.3, matrices M1, M2, and M3), the matrix multiplication algorithm (Algorithm 4) starts by multiplying M1 and M2, then the resulting matrix M4 is multiplied by M3, and so on. The sequential multiplication of matrices guarantees a global context across all words within the sentence. It also guarantees the maximum context value while maintaining the order of the terms within the sentence. The order of words in a sentence is critical to better understand and disambiguate the sentence. Finally, starting with the latest produced matrix, the back-tracing algorithm traces back all senses that contributed to the maximum global context.

Back-Tracing Senses: The final step of the SCSMM algorithm is the back-tracing stage (Algorithm 2 line 7). In this stage, we identify the most contributing sense to the sentence's global context (Algorithm 5). Fig. 4.5 and 4.4 illustrate the back-tracing stage as

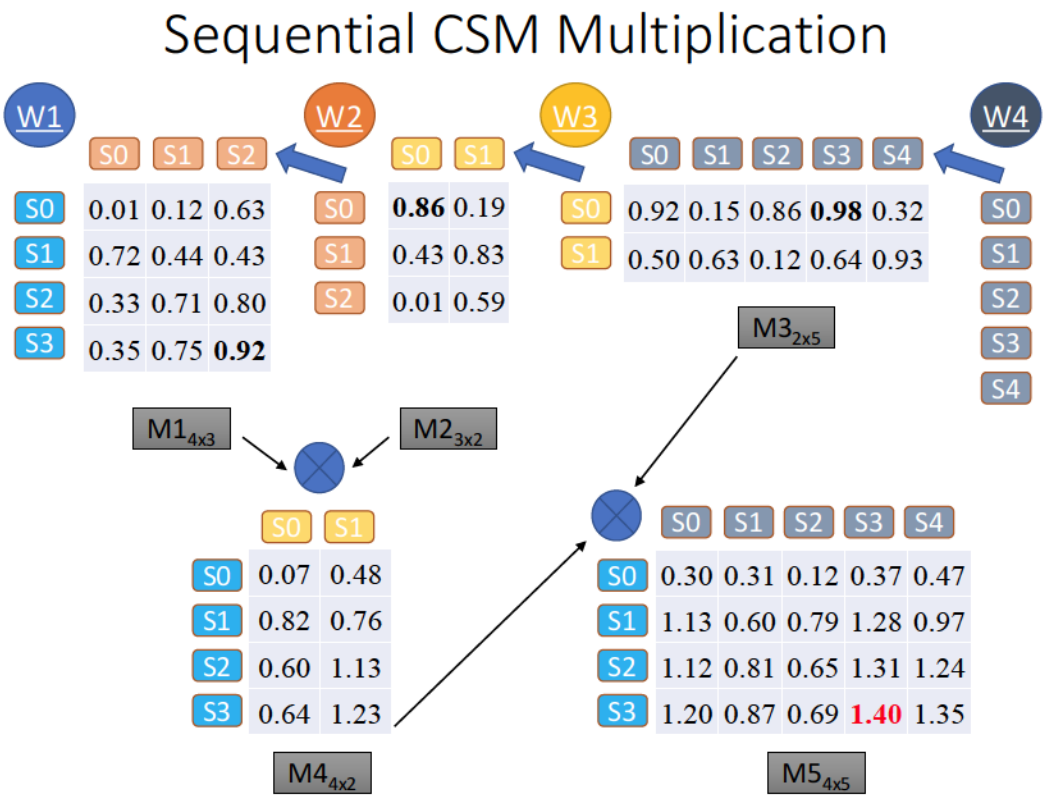


Figure 4.3: SCSMM illustration

follows: the back-tracing starts by selecting the maximum value from the final produced matrix. This value represents the maximum contextual weight for a given sentence. Then, this value is decomposed into its row and column vectors from the previous matrix multiplication. In step three, we select senses with the maximum product. These are senses that contributed the most to the global context. Finally, steps two and three repeat until there are no more elements to decompose.

As described above, our algorithm is intuitive and its results are explicable. It starts with a local context, then, it improves the context with heuristics and document context. Finally, it selects the most appropriate sense that contributes to the maximum global context while maintaining terms order.

4.3.2.3 Document Carry Forward Terms

In a few cases our algorithm is unable to disambiguate a term using the SCSMM algorithm. This would happen where a term has no local context (zero similarity) with its

Algorithm 5: Back-tracing the Maximum Context Contributing senses

Input : $MtxProductStack$: A Stack stores the product of the consecutive matricides

Output: $SensesList$: A stock of list of selected Senses

Data Structures:

Pr_{matrix}: Stores the previous matrix

Cr_{matrix}: Stores the current matrix

$location < r, c, val >$: triple <row, col, value> of the location of maximum value in the matrix

Initialization:

$Pr_{matrix} \xleftarrow{Pop} MtxProductStack$

$location\{r, c, val\} \leftarrow Max(Pr_{matrix})$

while $MtxProductStack \neq Empty$ **do**

  $SensesList \xleftarrow{Push} Sense(c)$

  $Cr_{matrix} \xleftarrow{Pop} MtxProductStack$

  /* The index of column that contribute the most to the context */

  $c \leftarrow Max(\{Row_{Cr}.Col_{Pr}\})$

  $location \leftarrow \{r, c, val\}$

  $Pr_{matrix} \leftarrow Cr_{matrix}$

$SensesList \xleftarrow{Push} Sense(c)$

$SensesList \xleftarrow{Push} Sense(r)$

return $SensesList$

surrounding terms. In such cases, we first attempt to disambiguate the term using its sentence as a context, including all recently disambiguated terms. Then we select the sense that has the maximum similarity with the sentence context. However, if a term still cannot be disambiguated within its own sentence, then, the term is carried forward to be disambiguated after the entire document is processed. These terms are referred to as Document Carry Forward (DocCF) terms. DocCF are processed after all sentences have been disambiguated to provide a maximum context for these terms. For each DocCF term, the sense

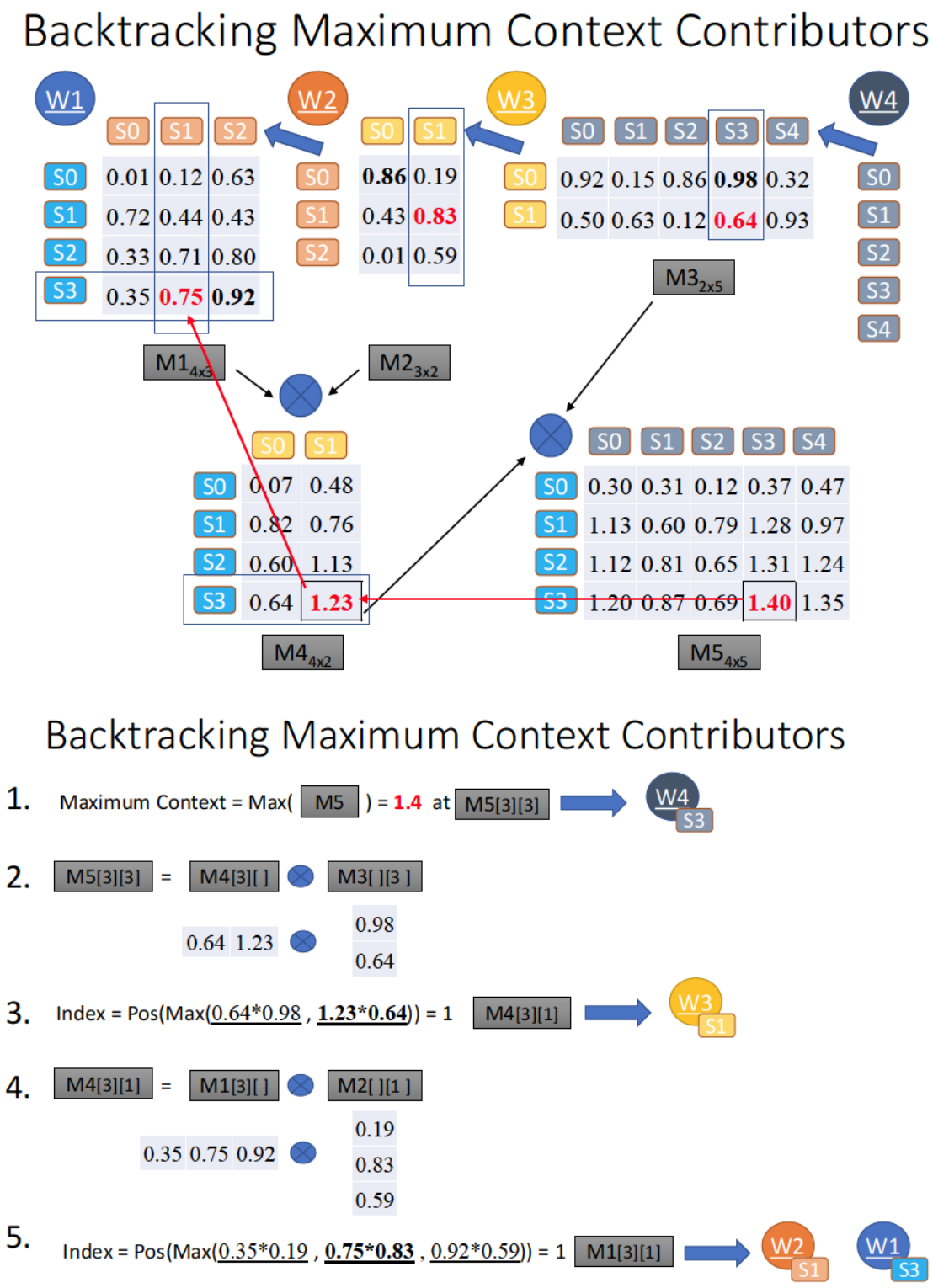


Figure 4.5: SCSMM back-tracing steps

with the maximum average similarity with all terms in the document is selected.

4.4 Evaluation and Experimental Results

4.4.1 Experimental Setup

We compared the results of our proposed SCSMM-WSD approach to the state-of-the-art systems based on well-known evaluation datasets. We also employed the commonly used training dataset in this field to obtain sense heuristic. We have also compared our approach to the baseline approaches represented by selecting the first sense in WordNet and the MFS using both training datasets. To obtain heuristics, we retrieved the senses' annotations from the SemCor and OMSTI training datasets (see Section 4.4.1.1). The SemCor annotations are available as part of the SemCor installation package in the 'cntlist' file, and the OMSTI annotations were preloaded to the SQL database from the 'keys' file downloaded from [123][2].

4.4.1.1 Training Datasets

The two large sense-annotated corpora (SemCor and OMSTI) have been used by many supervised approaches in training their models. Both datasets are tagged with WordNet senses. One of which is manually annotated, while the other is automatic.

- **SemCor** [14]: SemCor is a manually annotated corpus extracted from the original Brown corpus. The dataset is annotated with POS, lemmas, and word senses based on WordNet KG. SemCor consists of 352 documents: 186 documents include tags for all POS words (nouns, verbs, adjectives, and adverbs), while the remaining 166 contain tags only for verbs. The total number of sense annotations in all documents is 226,040. To our knowledge, SemCor is the largest manually annotated corpus with WordNet senses and is the main corpus used in various literature to train supervised WSD systems [103, 145].

- **OMSTI** [15]: OMSTI is an automatically sense-annotated corpus with senses from the WordNet 3.0. As the name suggests, it contains one million sense-annotated

[2] `http://lcl.uniroma1.it/wsdeval/home`

instances. To automatically tag senses, OMSTI used an English-Chinese parallel corpus[3] with an alignment-based WSD approach [16]. OMSTI has already shown its potential as a training corpus by improving the performance of supervised systems [15, 104].

4.4.1.2 Evaluation Datasets (Gold Standard)

A comprehensive evaluation framework has been presented in [123] with the integration of the primary WSD datasets. These datasets were presented as part of the SemEval International Workshop on Semantic Evaluation[4] between the years of 2002-2015. The framework included datasets from five main competitions, as presented in Table 4.3.

Table 4.3: SensEval/SemEval evaluation datasets

Dataset	Task	# of Senses				
Name	Method	NN	V	Adj	Adv	Total
SensEval2 (SE2) [146]	LS, AW	1066	517	445	254	2282
SensEval3 (SE3) [147]	LS, AW	900	588	350	12	1850
SemEval-07 (SE07) [148]	LS	159	296	-	-	455
SemEval-13 (SE13) [149]	LS, AW	1644	-	-	-	1644
SemEval-15 (SE15) [150]	LS, AW	531	251	160	80	1022

We further analyzed the datasets to determine the average sentence size, context size, and the ambiguity rate within each dataset. Table 4.4 depicts the statistics for each dataset. The average sentence size is calculated based on the number of annotated terms/processed sentence. Some sentences do not contain any terms; hence they are omitted. The context size is measured by the number of terms that have a single sense, hence unambiguous terms. Finally, the percentage of ambiguity is computed based on the number of ambiguous terms to the total number of terms. For example, *SemEval-07* has the highest ambiguity rate of 94%, with only 26 out of 455 terms that are not ambiguous (only one

[3] `http://www.euromatrixplus.net/multi-un/`

[4] Current workshop website: `http://alt.qcri.org/semeval2020/`

sense) and the smallest average sentence size with only, on average, three terms/sentence. Note that the ambiguity rate is inversely correlated with context size, which could degrade the disambiguation score, as presented in the results in Section 4.4.5.

Table 4.4: Statistics of WSD gold standard dataset

Criteria	SE2	SE3	SE07	SE13	SE15
#Doc	3	3	3	13	4
#Sent*	242	297	120	301	133
#Terms	2282	1850	455	1644	1022
AvgSentSize	9	6	3	5	7
Single sense	442	311	26	348	189
Ambiguity rate	81%	83%	94%	79%	82%

Furthermore, out of those ambiguous terms, Table 4.5 depicts the granularity level for each POS on all datasets combined. The granularity level reflects the average number of senses for each term. The granularity level negatively impacts disambiguation performance. Having a high granularity level makes the disambiguation decision very difficult even for humans, explaining the relatively low inter-agreement score between annotators. The annotators' inter-agreement score ranges between 72% to 80% on AW task. The average granularity level for verbs is the highest compared to all other POS; on average, each verb term has 10.95 senses compared to 5.71, 4.7, and 4.4 senses for the nouns, adjectives, and adverbs, respectively. The fourth row presents the maximum number of senses within each POS, where the maximum number of senses in verbs reaches up to 59, compared to 33, 21, and 13 senses for the nouns, adjectives, and adverbs, respectively. Both nouns and verbs are highly granular, explaining most systems' results as described later in Section 4.4.5. The mode and median also explain the results in Section 4.4.5, as most ambiguous verbs have four senses compared to two senses in all other POS.

Table 4.5: Ambiguous terms statistics for all gold standard datasets

	Noun	Verb	Adjective	Adverb
# of terms	4300	1652	955	346
# of ambiguous	3442	1555	732	208
Average granularity	5.7	11.0	4.7	4.4
Max #senses	33	59	21	13
Mode	2	4	2	2
Median	5	7	4	3

4.4.2 Evaluation Metric

Three main metrics are used to evaluate any WSD system performance: Precision, Recall, and F1-score. These measures are commonly used in the IR field. Assuming, within a dataset, there is a set of manually annotated test words $T = (w_1, ..., w_n)$, and for any system, the set of all evaluated/retrieved words is represented as $E = (w_1, ..., w_k) : k <= n$, and the set of correctly evaluated words $C = (w_1, ..., w_m) : m <= k$. Then we can evaluate the system as follow:

- **Precision:** the percentage of correctly identified words given by the system:

$$P = \frac{Number\,of\,correct\,words}{Number\,of\,evaluated\,words} = \frac{m}{k}, \tag{4}$$

  where $k = |E|$ the total number of evaluated words, and $m = |C|$ the total number of correctly evaluated words.

- **Recall:** the percentage of correctly identified words given by the system out of all test words in the dataset:

$$R = \frac{Number\,of\,Correct\,words}{Number\,of\,test\,words} = \frac{m}{n}, \tag{5}$$

  where $n = |T|$ the total number of evaluated words, and $m = |C|$ the total number of correctly evaluated words. If a system is able to evaluate every test word in T, then, we can say that the system has a 100% coverage, hence, $P = R$.

- **F1-score:** is a balanced F_{α}-$score$ where $\alpha = 0.5$. The F_1-$score$ is given by the following equation:

$$F_1\text{-}score = \frac{2PR}{P+R} \tag{6}$$

The general F_{α}-$score$ measures the trade-off between the precision and recall as follow:

$$F_{\alpha}\text{-}score = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} \tag{7}$$

4.4.3 Evaluated Semantic Similarity measures

In this section, we present various semantic similarity and relatedness measures that have been evaluated in our experiment. The similarity measure with the best performance is employed to construct the similarity matrix for our algorithm, as shown in Algorithm 3, Line 7. These measures have been discussed in detail in [119]. Table 4.6 depicts the performance of the top four measures (*LCH, WUP, JCN, and PATH*) on all dataset. As shown in these results, the JCN measure provides the best WSD performance across all datasets. The only exception is on *SemEval2013* (SE13) where both *PATH* and *LCH* outperformed *JCN*. However, using the combined datasets, *JCN* outperformed all other methods. Hence, it is the measure used in our SCSMM algorithm.

Table 4.6: F1-score for top four semantic similarity and relatedness methods

SSR	SE2	SE3	SE07	SE13	SE15	All
LCH	72.51	69.89	61.01	64.42	66.29	67.67
WUP	73.17	68.78	62.89	63.56	66.48	67.37
JCN	**78.14**	**72.67**	**64.78**	63.44	**68.38**	**69.67**
PATH	73.17	70.11	61.01	**64.66**	66.29	67.98

It is worth pointing that we have evaluated our PR-SSR from the first task with a small subset of the SemCor dataset (30 sentences). However, due to the WordNet version difference between the first and second tasks' implementations, we had run into a manual mapping of concepts from WordNet 3.0 (in the second task) and WordNet 3.1 (in the first

task). Therefore, to focus on the WSD algorithm, we applied the SSR measures that are part of the WordNet::Similarity library [143]. Aside from the implementation limitation, there are no theoretical blocks to prevent us from exploiting the benefits of the proposed PR-SSR to improve the SCSMM algorithm. As pointed in the last chapter, complete integration of the two tasks is part of the future works which would likely benefit the WSD system.

4.4.4 Implementation

Fig. 4.6 describes the architecture for the proposed WSD system. The WSD system is built based on the Web API architecture, which includes controllers and models. We further extend the architecture to provide separate services component that handles the main WSD system logic. The architecture consists of two Web API systems: the **WSD API** and the **PyNLTK API**. The **WSD API** is responsible for the core WSD algorithm, while the **PyNLTK API** is responsible for any NLP processing tasks, including the gloss-based similarity (i.e., Lesk).

The main WSD application is a C# Web API application with three separate layers: controllers, services, and the models. The controllers handle the API routing process and trigger the appropriate system logic from the services layer. In return, the services component is responsible for implementing the core WSD algorithm. It also connects with the models to add, retrieve, and update data from the database. Furthermore, the services layer is also responsible for establishing any internal or external API calls such as the calls to the **PyNLTK API** to perform any NLP pre-processing required or the calls the **BabelNet API**[5] to obtain BabeleNet synsets, which is required for the NSARI embedding evaluation.

The second **PyNLTK API** application is a python-based implementation. The main responsibility of this component is to compute text-based similarity measures, such as the *LESK* similarity.

[5] `https://babelnet.org/`

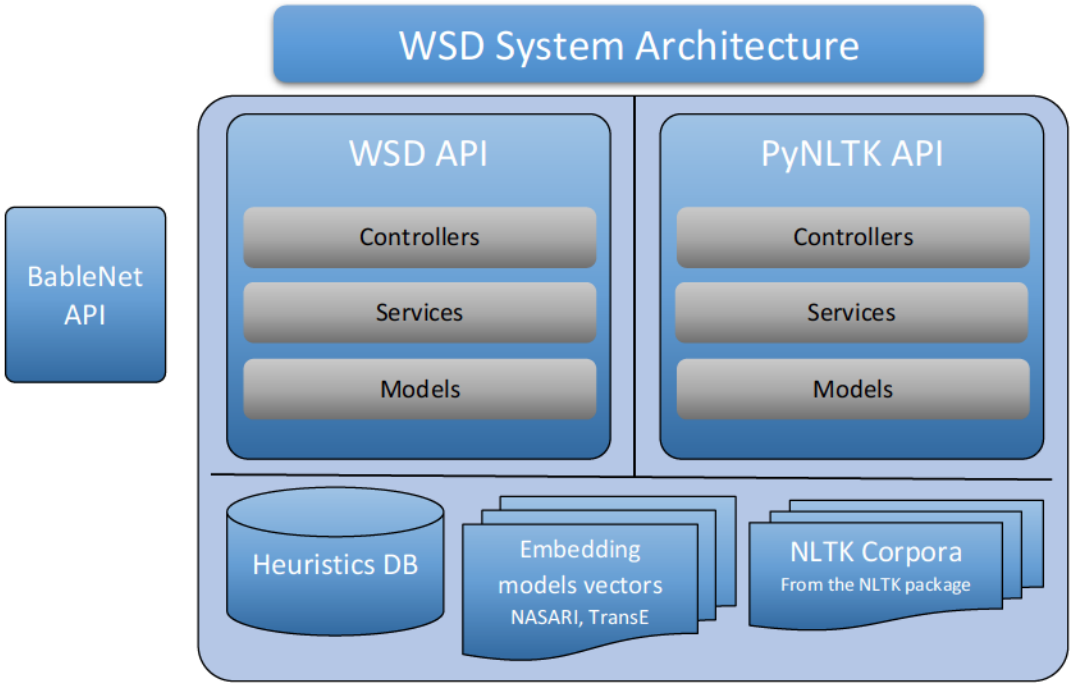


Figure 4.6: WSD system architecture

The data is retrieved from three distinct sources. The first is an SQL server database that stores the heuristics datasets (SemCor and OMSTI). The second consists of filesystems that contains pre-calculated embedding vectors for WordNet KG from two embedding models: NASARI [57], and TransE from [71]. The final is the NLTK corpora as part of the NLTK[6] package. We employed the Brown and SemCor corpora to compute concepts' IC.

Table 4.7 depicts the main parameters that control our system, where the right most column shows the optimal configuration that leads to the best performance. Note that we include all POS in the evaluation for the POS_OF_Int. However, since adjectives and adverbs merely describe the nouns and the verbs, respectively, they are not considered a context in DocCtxPOS parameter.

[6] `https://www.nltk.org/`

Table 4.7: Configuration parameters for the SCSMM system

Name	Description	Best Config.
SSR	Semantic similarity and relatedness measure	JCN
H(x)	Heuristic dataset used s=SemCor, so=SemCor+OMSTI	H(s)
DocCtx	Document Context used in the CSM flag	True
DocCF	Document carry forward flag	True
POS_OF_Int	The list of POS of interest that are being processed	{n,v,adj,adv}
DocCtxPOS	The list of POS used as in the document context	{n,v}

4.4.5 Experimental Results and Performance Analysis

To validate the robustness of the proposed method, we evaluated its performance with the five gold standard datasets presented in Table 4.3. We further present the results of the combined datasets to demonstrate the overall performance of the evaluated systems. The performance is measured by the F1-score discussed in Section 4.4.2. We present the proposed SCSMM method using two heuristics deployments; the first uses heuristics from the SemCor dataset (H_s), and the second uses both SemCor and OMSTI datasets[7] (H_{so}). In addition, we present three additional configurations for the SCSMM algorithm. These configurations demonstrate the effects of the document context and document carry forward on the performance of the proposed algorithm.

Table 4.8 depicts the F1-score for each individual dataset in addition to the overall performance on all five datasets combined. The results of all configurations of the proposed SCSMM algorithm are compared to the current state-of-the-art knowledge-based systems presented in [28, 29, 128, 131]. In addition, we present the baseline approaches using *WN1st sense*, the MFS_s, and the MFS_{so}.

The proposed SCSMM algorithm has the best performance when the document context is included in the CSM, and when the DocCF disambiguation option is enabled. SC-

[7]The training dataset were downloaded from `http://lcl.uniroma1.it/wsdeval/training-data`

Table 4.8: F1-score for each gold standard datasets

System	SE2	SE3	SE07	SE13	SE15	All
Lesk_{ext}	50.6	44.5	32.0	53.6	51.0	48.7
$\text{Lesk}_{\text{ext+emb}}$	63.0	63.7	56.7	66.2	64.6	63.7
UKB	56.0	51.7	39.0	53.6	55.2	53.2
$\text{UKB}_{\text{gloss}}$	60.6	54.1	42.0	59.0	61.2	57.5
Babelfy	67.0	63.5	51.6	66.4	**70.3**	65.5
$\text{UKB}_{\text{gloss18}}$	68.8	66.1	53.0	**68.8**	**70.3**	**67.3**
WSD-TM	**69.0**	66.9	55.6	65.3	69.6	66.9
WN1^{st} sense	66.8	66.2	55.2	63.0	67.8	65.2
MFS_{s}	65.6	66	54.5	63.8	67.1	64.8
MFS_{so}	66.5	60.4	52.3	62.6	64.2	62.8
$\text{SCSMM}_{H_{so}}$	66.9	67.2	55.4	63.0	68.4	65.6
SCSMM_{H_s}	68.1	67.2	55.4	63.0	68.4	66.0
$\text{SCSMM}_{H_s + DocCtx}$	68.4	66.8	56.9	63.4	69.0	66.2
$\text{SCSMM}_{H_s + DocCF}$	68.1	67.1	56.3	63.0	68.7	66.0
$\text{SCSMM}_{H_s + DocCtx + DocCF}$	68.9	**67.6**	**57.1**	63.5	69.5	66.7

SMM outperforms all other systems on two datasets, the SE3 and SE07, while matches the WSD-TM system on SE2. We noticed that our system is outperformed on SE13, as it is ranked fifth compared to other systems on the same datasets. We believe this is due to the following reasons: (1) This dataset is not diverse, as it only includes nouns, while we noticed with other datasets; that various POS contribute positively to the overall disambiguation algorithm. However, we could not prove this causation due to the effects of other factors and the limited datasets. (2) The other important factor is the average sentence size, as shown in Table 4.4. SE13 has an average sentence size of five terms per sentence, which is considered a small sentence size compared to other datasets. The only dataset that falls below that is SE07, which is explained next.

Finally, the SE07 dataset has shown a consistent drop in performance across all sys-

tems. According to our analysis, this drop is due to three main reasons. First, the high percentage of verbs within this dataset, as verbs have a very high granularity level that has an inverse proportional effect on the disambiguation score (see Table 4.5 and Fig. 4.8). Second, the dataset's small context size, as the entire dataset contains two nouns and 24 verbs as a context, making the SE07 dataset the most ambiguous dataset with a 96% ambiguity rate (see Table 4.4). Third and most importantly, the average sentence size. This dataset has the smallest average sentence size of three terms per sentence compared to all other datasets. Such a small average sentence size negatively impacts our algorithm because it identifies the global context between all terms, which is less accurate with smaller sentences.

Additionally, Table 4.9 depicts the F1-score of the combined five datasets on each POS. As can be seen from the results, our system outperforms all other systems when disambiguating nouns using the SCSMM$_{(H_s + DocCtx + DocCF)}$ with F1-score of 69.9. This is due to the proposed sequential algorithm that captures the maximum combination of the local similarities within each sentence. This can also be explained by the fact that nouns are structured and connected within WordNet compared to all other POS. Note that *Lesk$_{ext+emb}$ and WSD-TM* outperforms our system on verbs.

4.4.5.1 Discussion of Experimental Results

Despite the various scores achieved by the evaluated systems, Table 4.8 shows a performance correlation across all systems. The results demonstrate a consensus on the top and worst scores per dataset. For instance, most systems perform best on *SE15* and worst on *SE07*. Based on the observation above, we present and analyze the effect of POS distribution, granularity level, ambiguity rate, and sentence size on the performance of WSD systems in general and the proposed SCSMM algorithm in particular.

POS Distribution : The diversity of POS within each dataset appears to correlate with the F1-score. Fig. 4.7 depicts the F1-score for our proposed SCSMM algorithm with the POS distribution for each dataset. As shown in the figure, SE2 and SE15 contain similar

Table 4.9: F1-score for each POS on all gold standard datasets

System	Noun	Verb	Adj	Adv
Lesk_{ext}	54.1	27.9	54.6	60.3
$\text{Lesk}_{\text{ext+emb}}$	69.8	**51.2**	51.7	80.6
UKB	56.7	39.3	63.9	44.0
$\text{UKB}_{\text{gloss}}$	62.1	38.3	66.8	66.2
Babelfy	68.6	49.9	73.2	79.8
WSD-TM	69.7	**51.2**	**76.0**	**80.9**
WN1^{st} sense	67.6	50.3	74.3	**80.9**
MFS_{s}	67.6	49.6	73.1	80.5
MFS_{so}	65.8	45.9	72.7	80.5
$\text{SCSMM}_{H_{so}}$	68.2	50.5	74.6	80.1
SCSMM_{H_s}	68.9	50.5	74.7	80.1
$\text{SCSMM}_{H_s + DocCtx}$	69.8	50.1	73.6	78.6
$\text{SCSMM}_{H_s + DocCF}$	68.9	50.8	74.5	80.1
$\text{SCSMM}_{H_s + DocCtx + DocCF}$	**69.9**	51.0	74.7	80.3

POS distribution, in particular, the weights of verbs within the datasets has a higher impact on the performance of any WSD system, including the proposed algorithm. SE2 and SE15 contain almost the same percentage of verbs 23% and 25%, respectively, and have a very similar F1-score. As for the *SE3*, verbs occupy 32% of the dataset. Consequently, the performance of all systems has deteriorated for this dataset compared to *SE2* and *SE15*. Finally, having verbs outweigh nouns by almost double in *SE07*, all systems showed the lowest F1-score on this dataset compared to all other datasets.

Finally, the trigger dataset for analyzing the POS distribution is SE13. SE13 contains three of the best qualities a dataset could have, yet, it has low performance compared to other diverse datasets. SE13 contains only nouns, which are well structured in WordNet, it has the lowest ambiguity rate of 79% as shown in Table 4.4, and it has the lowest granularity level of 5.9 as a dataset (see Fig. 4.9). As a result, we conclude that a diverse

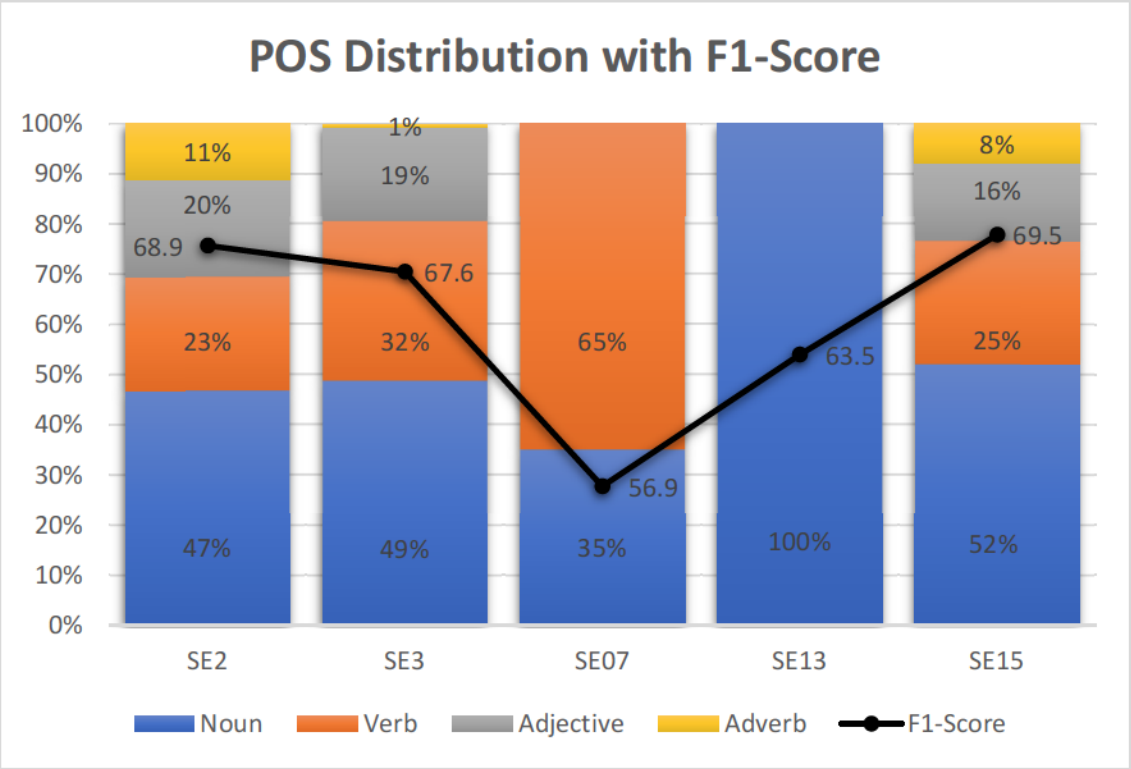


Figure 4.7: The distribution of POS compared to F1-score

distribution of POS within a dataset improves our WSD algorithm.

Granularity Level : Granularity level is one of the most apparent factors that affect the performance of any WSD system including the proposed algorithm. Fig. 4.8 exhibits the performance of the proposed system and all other evaluated systems compared to the granularity level for each POS. The columns in the figure represent the granularity levels, while the lines represent the F1-score for the evaluated systems. The figure clearly shows that the more granular senses within POS, the lower the system's performance. The same holds for the granularity level within each dataset regardless of the POS distribution. Fig. 4.9 presents the F1-score for all systems on each dataset compared to the granularity level of each dataset.

Context vs. Ambiguity Rates : Both *SE2* and *SE15* have almost the same POS distribution within their respective datasets (see Fig. 4.7) and the exact same granularity level

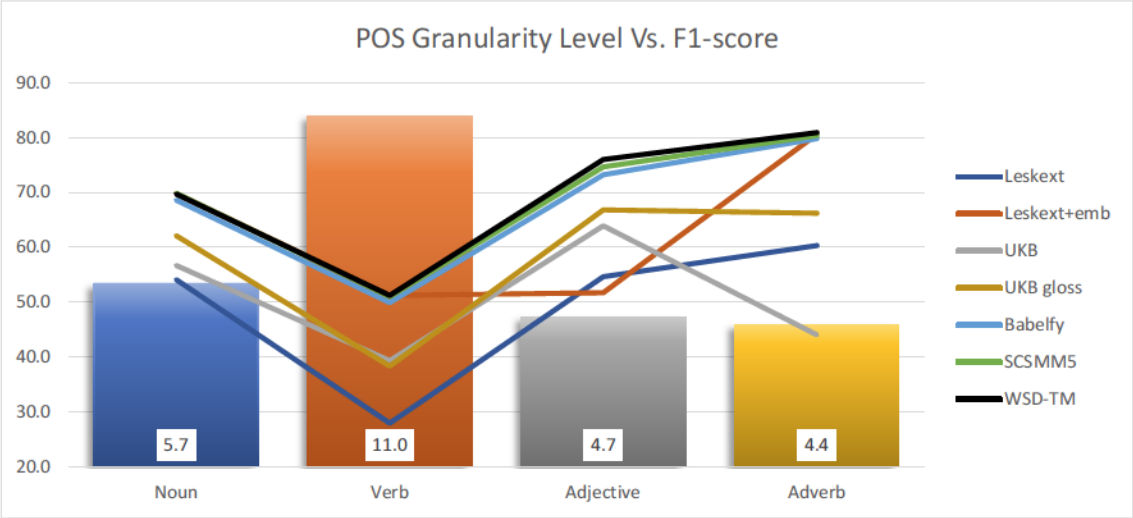


Figure 4.8: The granularity level of POS compared to F1-score

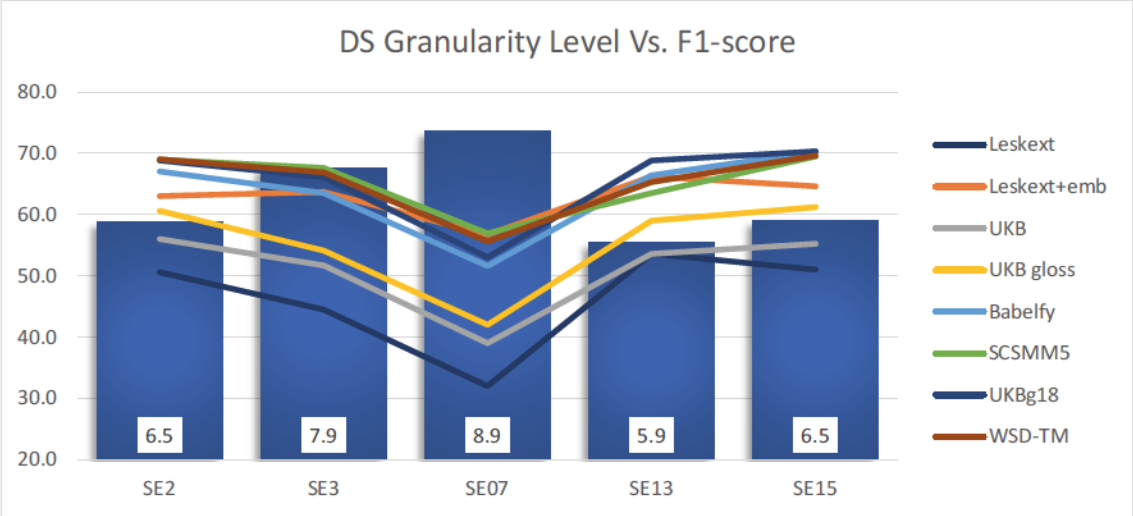


Figure 4.9: The granularity level of datasets compared to F1-score

(see Fig. 4.9). On the other hand, the other three datasets have different POS distribution and relatively higher granularity level. So what are the advantages of SE15 over SE2 that yield better performance? We believe this is due to the context and ambiguity rates. The ambiguity rate represents the percentage of ambiguous terms within each POS or dataset. Fig. 4.10 depicts the POS distribution for each dataset in addition to the context and ambiguity rates within each POS. Except for the nouns, SE15 has a higher context rate than SE2. This explains the results of the F1-score for each POS within these two datasets.

Table 4.10 shows the F1-scores for the proposed SCSMM$_{(H_s + DocCtx + DocCF)}$ algorithm for each POS on SE2 and SE15 datasets. The results correlate with the context and ambiguity rates within each POS. For example, SE2 has a higher context rate for the nouns than SE15; thus, it performed better. On the other hand, SE15 performed better than SE2 on all other POS due to their higher context rates.

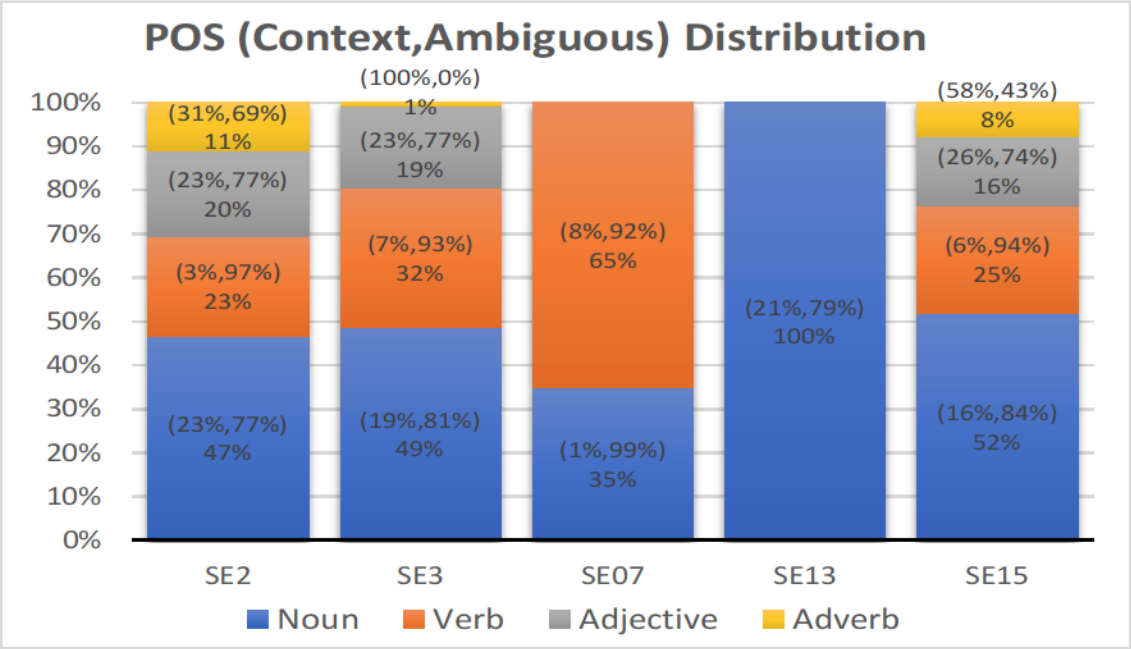


Figure 4.10: Distribution of POS with (context to ambiguous) ratio

Table 4.10: F1-score for SCSMM$_{(H_s + DocCtx + DocCF)}$ per POS

Dataset	Noun	Verb	Adjective	Adverb
SE2	77.5	43.3	73.0	78.0
SE15	69.5	57.4	80.6	85.0

Average Sentence Size : The average sentence size is the most important factor that affects the performance of our SCSMM algorithm and other systems, as it is more challenging to extract a context from fewer words. The same is true for a large number of words. The average sentence size is shown in Table 4.4, which explains the lower per-

formance of SE13 compared to SE2 as the average sentence size is smaller for SE13. However, although SE15 has a smaller average sentence size than SE2, it performed better. This result can be justified by the context rate factor discussed above, or an indication of an optimal average sentence size.

4.5 Conclusion

In this chapter, we presented a novel knowledge-based WSD approach. The proposed SC-SMM-WSD approach provides comparable results and outperforms most of the current state-of-the-art KG-based systems. Moreover, we evaluated the performance of current WSD systems, including our proposed method, on well-known gold standard datasets from the SemEval workshop series. Based on the datasets analysis and the trends of the evaluated systems' results, we conclude that WSD systems are impacted by the granularity level of the dataset and the included POS, the diversity of POS within the dataset, the context to ambiguity rate, and the average sentence size.

We believe that as the KGs are enriched with more relationships between entities, and more domain-based KG are exploited, knowledge base systems will outperform other WSD approaches. Furthermore, knowledge base systems are intuitive, and their results are easily explained, understood, and justified by a human.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This thesis presented a novel semantic similarity and relatedness measure, namely PR-SSR, that combined taxonomic and non-taxonomic relations. Our proposed measure is based on two new parameters that describe non-taxonomic relations. The first is a relation's IC (RIC) that quantifies the amount of information exchanged between the concepts at both edges of the relation. The second is the relation's prevalence. The prevalence measure reflects the importance and relevancy of a relation within the KG. We believe that the prevalence measure can show better results if a domain-specific KG is used.

The proposed semantic similarity and relatedness technique can be applied to many research domains such as IR, Semantic Recommender Systems, and NLP. For example, in social media, non-taxonomic relations are dominant and can be used to infer new insights about entities in the semantic graph (i.e., friends, places, products, and services). In a Recommender Systems for an e-store, products can be suggested to customers based on their semantic relations with other products. Similarity Search engines can benefit significantly from such measure, especially when there exists a semantically rich KG. However, the proposed semantic similarity and relatedness approach has a limitation in the sense that it cannot be applied to a knowledge graph with few or no non-taxonomic

relations (semantically-poor KG).

In Chapter 4, we proposed a novel knowledge-based WSD approach that follows the brain intuition in disambiguating words within a sentence. The proposed SCSMM algorithm identifies senses that provide the maximum global context within a sentence. Unlike other systems, our proposed SCSMM algorithm exploits the merits of local context, word sense heuristic, and the global context while maintaining the words order.

This thesis also presented a detailed analysis of the core factors that affect any WSD system. These factors include the diversity of POS distribution, the granularity level of the dataset, the granularity level of the POS within the dataset, the context to ambiguity rate, and the average sentence size. Based on the datasets analysis and the trends of the evaluated systems, we conclude that WSD systems are impacted negatively by the granularity level of the dataset and the included POS. On the other hand, a more diverse POS within the dataset improves the results of the proposed WSD algorithm. Similarly, the higher the context rate, the better F1-score. Finally, the results show that the proposed SCSMM algorithm can be negatively impacted by very short sentences (i.e., less than three words).

5.2 Future Work

Despite the improvement of the proposed PR-SSR measure, it has a major dependency on existing taxonomic IC-based measures denoted by the baselines presented in Section 3.2. This dependency is due to the usage of existing taxonomic-based similarity measures. As future work, these baselines can be eliminated to develop a new relatedness measure based on the relations-IC and prevalence of both taxonomic and non-taxonomic relations. Furthermore, such a new measure could positively impact the WSD algorithm with full integration of the two tasks, as it provides additional semantic representation between terms. Finally, it would be interesting to employ both algorithms (PR-SSR and SCSMM) to solve more complex problems such as semantic sentence similarity and topic detection. The semantic representation of concepts within the KG is an integral measure for

semantic-based WSD. Current knowledge-based WSD methods do not utilize this measure to its full potential. We believe that this is because the currently available KGs are limited in semantically representing many real-world relations between concepts. Hence, we need a domain-specific KG in addition to the current massive KG, which was beyond the scope of our research.

Another limitation of the proposed algorithms is the semantically-poor KG. This is due to the lack of domain specific non-taxonomic relations, which limits the contextualised connections between relations, hence the semantic similarity and relatedness value between terms. To overcome this limitation, a semantically rich domain-specific KG can be developed and incorporated to enhance both the semantic similarity and relatedness between terms and WSD results. Finally, the proposed SCSMM method does not capture the exact topic of the document, but rather utilizes all context words in the document to disambiguate terms. To address this limitation, future research could investigate the adaptation of topic modeling and text clustering algorithms such as the LDA algorithms used in [132].

Bibliography

[1] S. Soler and A. Montoyo, "A proposal for wsd using semantic similarity," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2002, pp. 165–167.

[2] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge acquisition*, vol. 5, no. 2, pp. 199–220, 1993.

[3] L. Ehrlinger and W. Wöß, "Towards a definition of knowledge graphs." *SEMANTiCS (Posters, Demos, SuCCESS)*, vol. 48, 2016.

[4] M. Barthélemy, E. Chow, and T. Eliassi-Rad, "Knowledge representation issues in semantic graphs for relationship detection." in *AAAI Spring Symposium: AI Technologies for Homeland Security*, vol. 3, 2005, pp. 91–98.

[5] T. Eliassi-Rad and E. Chow, "Using ontological information to accelerate path-finding in large semantic graphs: A probabilistic approach," in *American Association for Artificial Intelligence*, 2005.

[6] J. Hebeler, M. Fisher, R. Blace, and A. Perez-Lopez, *Semantic web programming*. John Wiley & Sons, 2011.

[7] D. Chandrasekaran and V. Mago, "Evolution of semantic similarity–a survey," *arXiv preprint arXiv:2004.13820*, 2020.

[8] Y. Cai, S. Pan, X. Wang, H. Chen, X. Cai, and M. Zuo, "Measuring distance-based semantic similarity using meronymy and hyponymy relations," *Neural Computing and Applications*, pp. 1–14, 2018.

[9] Y. Cai, Q. Zhang, W. Lu, and X. Che, "A hybrid approach for measuring semantic similarity based on ic-weighted path distance in wordnet," *Journal of Intelligent Information Systems*, vol. 51, no. 1, pp. 23–47, 2017.

[10] J. J. Lastra-Díaz and A. García-Serrano, "A new family of information content models with an experimental survey on wordnet," *Knowledge-Based Systems*, vol. 89, pp. 509–526, 2015.

[11] M. A. H. Taieb, M. B. Aouicha, and A. B. Hamadou, "Ontology-based approach for measuring semantic similarity," *Engineering Applications of Artificial Intelligence*, vol. 36, pp. 238–261, 2014.

[12] S. A. Elavarasi, J. Akilandeswari, and K. Menaga, "A survey on semantic similarity measure," *International Journal of Research in Advent Technology*, vol. 2, no. 3, pp. 389–398, 2014.

[13] J. J. Lastra-Díaz, J. Goikoetxea, M. A. H. Taieb, A. García-Serrano, M. B. Aouicha, and E. Agirre, "A reproducible survey on word embeddings and ontology-based methods for word similarity: linear combinations outperform the state of the art," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 645–665, 2019.

[14] G. A. Miller, M. Chodorow, S. Landes, C. Leacock, and R. G. Thomas, "Using a semantic concordance for sense identification," in *HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994.

[15] K. Taghipour and H. T. Ng, "One million sense-tagged instances for word sense disambiguation and induction," in *In Proceedings of the nineteenth conference on computational natural language learning*, 2015, pp. 338–344.

[16] Y. S. Chan and H. T. Ng, "Scaling up word sense disambiguation via parallel texts," in *AAAI*, vol. 5, 2005, pp. 1037–1042.

[17] S. Parameswarappa, V. Narayana, and D. Yarowsky, "Kannada word sense disambiguation using decision list," *Volume*, vol. 2, pp. 272–278, 2013.

[18] R. J. Mooney, "Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning," *arXiv preprint cmp-lg/9612001*, 1996.

[19] G. Escudero, L. Màrquez, G. Rigau, and J. G. Salgado, "On the portability and tuning of supervised word sense disambiguation systems," in *In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. Citeseer, 2000.

[20] L. Vial, B. Lecouteux, and D. Schwab, "Improving the coverage and the generalization ability of neural word sense disambiguation through hypernymy and hyponymy relationships," *arXiv preprint arXiv:1811.00960*, 2018.

[21] ——, "Sense vocabulary compression through the semantic knowledge of wordnet for neural word sense disambiguation," *arXiv preprint arXiv:1905.05677*, 2019.

[22] G. Wiedemann, S. Remus, A. Chawla, and C. Biemann, "Does bert make any sense? interpretable word sense disambiguation with contextualized embeddings," *arXiv preprint arXiv:1909.10430*, 2019.

[23] A. Raganato, C. D. Bovi, and R. Navigli, "Neural sequence learning models for word sense disambiguation," in *In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 1156–1167.

[24] O. Melamud, J. Goldberger, and I. Dagan, "context2vec: Learning generic context embedding with bidirectional lstm," in *In Proceedings of the 20th SIGNLL conference on computational natural language learning*, 2016, pp. 51–61.

[25] R. Mihalcea and E. Faruque, "Senselearner: Minimally supervised word sense disambiguation for all words in open text," in *In Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 2004, pp. 155–158.

[26] H. Ji, "One sense per context cluster: Improving word sense disambiguation using

web-scale phrase clustering," in *2010 4th International Universal Communication Symposium*. IEEE, 2010, pp. 181–184.

[27] P. Pantel and D. Lin, "Discovering word senses from text," in *In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 613–619.

[28] E. Agirre and A. Soroa, "Personalizing pagerank for word sense disambiguation," in *In Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, 2009, pp. 33–41.

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

[30] S. Banerjee and T. Pedersen, "Extended gloss overlaps as a measure of semantic relatedness," in *Ijcai*, vol. 3, 2003, pp. 805–810.

[31] H. Liu, H. Bao, and D. Xu, "Concept vector for semantic similarity and relatedness based on wordnet structure," *Journal of Systems and software*, vol. 85, no. 2, pp. 370–381, 2012.

[32] N. Seco, T. Veale, and J. Hayes, "An intrinsic information content metric for semantic similarity in wordnet," in *Ecai*, vol. 16, 2004, p. 1089.

[33] A. Sebti and A. A. Barfroush, "A new word sense similarity measure in wordnet," in *2008 International Multiconference on Computer Science and Information Technology*. IEEE, 2008, pp. 369–373.

[34] D. Sánchez, M. Batet, and D. Isern, "Ontology-based information content computation," *Knowledge-Based Systems*, vol. 24, no. 2, pp. 297–303, 2011.

[35] Z. Zhou, Y. Wang, and J. Gu, "New model of semantic similarity measuring in wordnet," in *2008 3rd International Conference on Intelligent System and Knowledge Engineering*, vol. 1. IEEE, 2008, pp. 256–261.

[36] Daniel Jurafsky and James H. Martin, "Chapter 19: Word senses and wordnet," in *Speech and Language Processing*. Third Edition draft, 2018.

[37] R. Navigli, "Word sense disambiguation: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 2, pp. 1–69, 2009.

[38] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "Dbpedia-a crystallization point for the web of data," *Web Semantics: science, services and agents on the world wide web*, vol. 7, no. 3, pp. 154–165, 2009.

[39] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *In SIGMOD Conference*, 2008, pp. 1247–1250.

[40] F. Mahdisoltani, J. Biega, and F. M. Suchanek, "Yago3: A knowledge base from multilingual wikipedias," in *CIDR 2015 - 7th Biennial Conference on Innovative Data Systems Research*, 2013.

[41] R. Navigli and S. P. Ponzetto, "Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network," *Artif. Intell.*, vol. 193, p. 217–250, Dec. 2012.

[42] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[43] Z. Zhou, Y. Wang, and J. Gu, "A new model of information content for semantic similarity in wordnet," in *2008 Second International Conference on Future Generation Communication and Networking Symposia*, vol. 3. IEEE, 2008, pp. 85–89.

[44] S. Kim, N. Fiorini, W. J. Wilbur, and Z. Lu, "Bridging the gap: incorporating a semantic similarity measure for effectively mapping pubmed queries to documents," *Journal of biomedical informatics*, vol. 75, pp. 122–127, 2017.

[45] T. Wei, Y. Lu, H. Chang, Q. Zhou, and X. Bao, "A semantic approach for text clustering using wordnet and lexical chains," *Expert Systems with Applications*, vol. 42, no. 4, pp. 2264–2275, 2015.

[46] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[47] M. Mohamed and M. Oussalah, "Srl-esa-textsum: A text summarization approach based on semantic role labeling and explicit semantic analysis," *Information Processing & Management*, vol. 56, no. 4, pp. 1356–1372, 2019.

[48] T. Miller, C. Biemann, T. Zesch, and I. Gurevych, "Using distributional similarity for lexical expansion in knowledge-based word sense disambiguation," in *In Proceedings of COLING 2012*, 2012, pp. 1781–1796.

[49] X. Wang, Y. Liu, and F. Xiong, "Improved personalized recommendation based on a similarity network," *Physica A: Statistical Mechanics and its Applications*, vol. 456, pp. 271–280, 2018.

[50] K. Janowicz, M. Raubal, and W. Kuhn, "The semantics of similarity in geographic information retrieval," *Journal of Spatial Information Science*, vol. 2011, no. 2, pp. 29–57, 2011.

[51] P. H. Guzzi, M. Mina, C. Guerra, and M. Cannataro, "Semantic similarity analysis of protein data: assessment with biological features and issues," *Briefings in bioinformatics*, vol. 13, no. 5, pp. 569–585, 2012.

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

[53] Z. Wang, H. Mi, and A. Ittycheriah, "Sentence similarity learning by lexical decomposition and composition," *arXiv preprint arXiv:1602.07019*, 2016.

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

[55] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *In Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, 2013, pp. 746–751.

[56] H. He and J. Lin, "Pairwise word interaction modeling with deep neural networks for semantic similarity measurement," in *In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 937–948.

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

[58] G. Zhu and C. A. Iglesias, "Computing semantic similarity of concepts in knowledge graphs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 72–85, 2016.

[59] R. Rada, H. Mili, E. Bicknell, and M. Blettner, "Development and application of a metric on semantic nets," *IEEE transactions on systems, man, and cybernetics*, vol. 19, no. 1, pp. 17–30, 1989.

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

[61] Y. Li, Z. A. Bandar, and D. McLean, "An approach for measuring semantic similarity between words using multiple information sources," *IEEE Transactions on knowledge and data engineering*, vol. 15, no. 4, pp. 871–882, 2003.

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

[63] D. Lin, "An information-theoretic definition of similarity." in *Icml*, vol. 98. Citeseer, 1998, pp. 296–304.

[64] X. Zhang, S. Sun, and K. Zhang, "An information content-based approach for measuring concept semantic similarity in wordnet," *Wireless Personal Communications*, vol. 103, no. 1, pp. 117–132, 2018.

[65] L. Meng, J. Gu, and Z. Zhou, "A new model of information content based on concept's topology for measuring semantic similarity in wordnet," *International Journal of Grid and Distributed Computing*, vol. 5, no. 3, pp. 81–94, 2012.

[66] A. Tversky, "Features of similarity." *Psychological review*, vol. 84, no. 4, p. 327, 1977.

[67] G. Zhu and C. A. Iglesias, "Sematch: Semantic similarity framework for knowledge graphs," *Knowledge-Based Systems*, vol. 130, pp. 30–32, 2017.

[68] D. Sánchez, M. Batet, D. Isern, and A. Valls, "Ontology-based semantic similarity: A new feature-based approach," *Expert systems with applications*, vol. 39, no. 9, pp. 7718–7728, 2012.

[69] C. Yang, Y. Zhu, M. Zhong, and R. Li, "Semantic similarity computation in knowledge graphs: Comparisons and improvements," in *2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)*. IEEE, 2019, pp. 249–252.

[70] Y. Jiang, X. Zhang, Y. Tang, and R. Nie, "Feature-based approaches to semantic similarity assessment of concepts using wikipedia," *Information Processing & Management*, vol. 51, no. 3, pp. 215–234, 2015.

[71] S. Y. Yu, S. R. Chhetri, A. Canedo, P. Goyal, and M. A. A. Faruque, "Pykg2vec: A python library for knowledge graph embedding," *arXiv preprint arXiv:1906.04239*, 2019.

[72] G. Pirró, "A semantic similarity metric combining features and intrinsic information content," *Data & Knowledge Engineering*, vol. 68, no. 11, pp. 1289–1308, 2009.

[73] M. J. Hussain, S. H. Wasti, G. Huang, L. Wei, Y. Jiang, and Y. Tang, "An approach for measuring semantic similarity between wikipedia concepts using multiple inheritances," *Information Processing and Management*, vol. 57, no. 3, p. 102188, 2020.

[74] S. Patwardhan and T. Pedersen, "Using wordnet-based context vectors to estimate the semantic relatedness of concepts," in *In Proceedings of the Workshop on Making Sense of Sense: Bringing Psycholinguistics and Computational Linguistics Together*, 2006.

[75] A. Pesaranghader, S. Muthaiyah, and A. Pesaranghader, "Improving gloss vector semantic relatedness measure by integrating pointwise mutual information: Optimizing second-order co-occurrence vectors computed from biomedical corpus and umls," in *2013 International Conference on Informatics and Creative Multimedia*. IEEE, 2013, pp. 196–201.

[76] T. Chklovski and R. Mihalcea, "Exploiting agreement and disagreement of human annotators for word sense disambiguation," in *In Proceedings of Recent Advances In NLP (RANLP 2003)*, 2003.

[77] M. Palmer, H. T. Dang, and C. Fellbaum, "Making fine-grained and coarse-grained sense distinctions, both manually and automatically," *Nat. Lang. Eng.*, vol. 13, no. 2, pp. 137–163, 2007.

[78] B. Snyder and M. Palmer, "The english all-words task," in *In Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 2004, pp. 41–43.

[79] C. Lacerra, M. Bevilacqua, T. Pasini, and R. Navigli, "Csi: A coarse sense inventory for 85% word sense disambiguation." in *AAAI*, 2020, pp. 8123–8130.

[80] W. A. Gale, K. Church, and D. Yarowsky, "Estimating upper and lower bounds on the performance of word-sense disambiguation programs," in *30th Annual Meeting of the Association for Computational Linguistics*, 1992, pp. 249–256.

[81] A. R. Gonzales, L. Mascarell, and R. Sennrich, "Improving word sense disambiguation in neural machine translation with sense embeddings," in *In Proceedings of the Second Conference on Machine Translation*, 2017, pp. 11–19.

[82] D. Xiong and M. Zhang, "A sense-based translation model for statistical machine translation," in *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 1459–1469.

[83] F. D. Paskalis and M. L. Khodra, "Word sense disambiguation in information retrieval using query expansion," in *In Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*. IEEE, 2011, pp. 1–6.

[84] C. Stokoe, M. P. Oakes, and J. Tait, "Word sense disambiguation in information retrieval revisited," in *In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2003, pp. 159–166.

[85] J. C. Hung, C.-S. Wang, C.-Y. Yang, M.-S. Chiu, and G. Yee, "Applying word sense disambiguation to question answering system for e-learning," in *19th International Conference on Advanced Information Networking and Applications (AINA'05) Volume 1 (AINA papers)*, vol. 1. IEEE, 2005, pp. 157–162.

[86] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: a simple and general method for semi-supervised learning," in *In Proceedings of the 48th annual meeting of the association for computational linguistics*, 2010, pp. 384–394.

[87] H. Liu, Y. A. Lussier, and C. Friedman, "Disambiguating ambiguous biomedical terms in biomedical narrative text: an unsupervised method," *Journal of biomedical informatics*, vol. 34, no. 4, pp. 249–261, 2001.

[88] P. P. Borah, G. Talukdar, and A. Baruah, "Approaches for word sense disambiguation–a survey," *International Journal of Recent Technology and Engineering*, vol. 3, no. 1, pp. 35–38, 2014.

[89] A. R. Pal and D. Saha, "Word sense disambiguation: A survey," *arXiv preprint arXiv:1508.01346*, 2015.

[90] R. Giyanani, "A survey on word sense disambiguation," *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol. 14, pp. 30–33, 2013.

[91] J. Sarmah and S. K. Sarma, "Survey on word sense disambiguation: an initiative towards an indo-aryan language," *IJEM*, vol. 6, no. 3, pp. 37–52, 2016.

[92] A. H. Aliwy and H. A. Taher, "Word sense disambiguation: Survey study," *Journal of Computer Science*, vol. 15, no. 7, pp. 1004–1011, July 2019. [Online]. Available: http://thescipub.com/abstract/10.3844/jcssp.2019.1004.1011

[93] T. Pasini and R. Navigli, "Train-o-matic: Supervised word sense disambiguation with no (manual) effort," *Artificial Intelligence*, vol. 279, p. 103215, 2020.

[94] R. L. Rivest, "Learning decision lists," *Machine learning*, vol. 2, no. 3, pp. 229–246, 1987.

[95] D. Yarowsky, "Hierarchical decision lists for word sense disambiguation," *Computers and the Humanities*, vol. 34, no. 1-2, pp. 179–186, 2000.

[96] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[97] ——, *C4.5: programs for machine learning*. Elsevier, 1993.

[98] H. T. Ng, "Getting serious about word sense disambiguation," in *Tagging Text with Lexical Semantics: Why, What, and How?*, 1997.

[99] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *In Proceedings of the fifth annual workshop on Computational learning theory*, 1992, pp. 144–152.

[100] Y. K. Lee, H. T. Ng, and T. K. Chia, "Supervised word sense disambiguation with support vector machines and multiple knowledge sources," in *In Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 2004, pp. 137–140.

[101] M. Joshi, T. Pedersen, and R. Maclin, "A comparative study of support vector machines applied to the supervised word sense disambiguation problem in the medical domain." in *IICAI*, 2005, pp. 3449–3468.

[102] D. Buscaldi, P. Rosso, F. Pla, E. Segarra, and E. S. Arnal, "Verb sense disambiguation using support vector machines: Impact of wordnet-extracted features," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2006, pp. 192–195.

[103] Z. Zhong and H. T. Ng, "It makes sense: A wide-coverage word sense disambiguation system for free text," in *In Proceedings of the ACL 2010 system demonstrations*, 2010, pp. 78–83.

[104] I. Iacobacci, M. T. Pilehvar, and R. Navigli, "Embeddings for word sense disambiguation: An evaluation study," in *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 897–907.

[105] S. Papandrea, A. Raganato, and C. D. Bovi, "Supwsd: A flexible toolkit for supervised word sense disambiguation," in *In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2017, pp. 103–108.

[106] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[107] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm networks," in *In Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 4. IEEE, 2005, pp. 2047–2052.

[108] H. T. Ng and H. B. Lee, "Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach," *arXiv preprint cmp-lg/9606032*, 1996.

[109] D. Yuan, J. Richardson, R. Doherty, C. Evans, and E. Altendorf, "Semi-supervised word sense disambiguation with neural models," *arXiv preprint arXiv:1603.07012*, 2016.

[110] H. Schütze, "Dimensions of meaning," in *Supercomputing'92: Proceedings of the 1992 ACM/IEEE Conference on Supercomputing*. IEEE, 1992, pp. 787–796.

[111] ——, "Automatic word sense discrimination," *Computational linguistics*, vol. 24, no. 1, pp. 97–123, 1998.

[112] T. Pedersen and R. Bruce, "Distinguishing word senses in untagged text," *arXiv preprint cmp-lg/9706008*, 1997.

[113] B. Dorow and D. Widdows, "Discovering corpus-specific word senses," in *10th Conference of the European Chapter of the Association for Computational Linguistics*, 2003.

[114] D. Widdows and B. Dorow, "A graph model for unsupervised lexical acquisition," in *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

[115] J. Véronis, "Hyperlex: lexical cartography for information retrieval," *Computer Speech & Language*, vol. 18, no. 3, pp. 223–252, 2004.

[116] E. Agirre and M. Stevenson, "Knowledge sources for wsd," in *Word Sense Disambiguation*. Springer, 2007, pp. 217–251.

[117] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," in *In Proceedings of the Seventh International Conference on World Wide Web 7*, ser. WWW7. NLD: Elsevier Science Publishers B. V., 1998, p. 107–117.

[118] D. Lin, "Automatic retrieval and clustering of similar words," in *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, 1998, pp. 768–774.

[119] M. AlMousa, R. Benlamri, and R. Khoury, "Exploiting non-taxonomic relations for measuring semantic similarity and relatedness in wordnet," *arXiv preprint arXiv:2006.12106*, 2020.

[120] T. Pedersen, S. Banerjee, and S. Patwardhan, "Maximizing semantic relatedness to perform word sense disambiguation," Research Report UMSI 2005/25, University of Minnesota Supercomputing Institute, Tech. Rep., 2005.

[121] K. Mittal and A. Jain, "Word sense disambiguation method using semantic similarity measures and owa operator," *ICTACT Journal on Soft Computing*, vol. 5, no. 2, 2015.

[122] C. Leacock and M. Chodorow, "Combining local context and wordnet similarity for word sense identification," *WordNet: An electronic lexical database*, vol. 49, no. 2, pp. 265–283, 1998.

[123] A. Raganato, J. Camacho-Collados, and R. Navigli, "Word sense disambiguation: A unified evaluation framework and empirical comparison," in *In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 2017, pp. 99–110.

[124] R. Navigli and M. Lapata, "Graph connectivity measures for unsupervised word sense disambiguation," in *IJCAI*, vol. 7, 2007, pp. 1683–1688.

[125] O. Dongsuk, S. Kwon, K. Kim, and Y. Ko, "Word sense disambiguation based on word similarity calculation using word vector representation from a knowledge-based graph," in *In Proceedings of the 27th international conference on computational linguistics*, 2018, pp. 2704–2714.

[126] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, 2014, pp. 1188–1196.

[127] P. Basile, A. Caputo, and G. Semeraro, "An enhanced lesk word sense disambiguation algorithm through a distributional semantic model," in *In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 1591–1600.

[128] E. Agirre, O. López de Lacalle, and A. Soroa, "Random walks for knowledge-based word sense disambiguation," *Computational Linguistics*, vol. 40, no. 1, pp. 57–84, 2014.

[129] E. Agirre, O. L. de Lacalle, and A. Soroa, "The risk of sub-optimal use of open

source nlp software: Ukb is inadvertently state-of-the-art in knowledge-based wsd," *arXiv preprint arXiv:1805.04277*, 2018.

[130] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," in *Sixth international conference on data mining (ICDM'06)*. IEEE, 2006, pp. 613–622.

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

[132] D. S. Chaplot and R. Salakhutdinov, "Knowledge-based word sense disambiguation using topic models," *arXiv preprint arXiv:1801.01900*, 2018.

[133] R. Mihalcea, "Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling," in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 2005, pp. 411–418.

[134] R. Sinha and R. Mihalcea, "Unsupervised graph-basedword sense disambiguation using measures of word semantic similarity," in *International conference on semantic computing (ICSC 2007)*. IEEE, 2007, pp. 363–369.

[135] K. Patterson, P. J. Nestor, and T. T. Rogers, "Where do you know what you know? the representation of semantic knowledge in the human brain," *Nature reviews neuroscience*, vol. 8, no. 12, pp. 976–987, 2007.

[136] X. Liao, A. V. Vasilakos, and Y. He, "Small-world human brain networks: perspectives and challenges," *Neuroscience & Biobehavioral Reviews*, vol. 77, pp. 286–300, 2017.

[137] M. P. van den Heuvel and O. Sporns, "Network hubs in the human brain," *Trends in cognitive sciences*, vol. 17, no. 12, pp. 683–696, 2013.

[138] M. Xia, J. Wang, and Y. He, "Brainnet viewer: a network visualization tool for human brain connectomics," *PloS one*, vol. 8, no. 7, p. e68910, 2013.

[139] G. A. Miller and W. G. Charles, "Contextual correlates of semantic similarity," *Language and cognitive processes*, vol. 6, no. 1, pp. 1–28, 1991.

[140] H. Rubenstein and J. B. Goodenough, "Contextual correlates of synonymy," *Communications of the ACM*, vol. 8, no. 10, pp. 627–633, 1965.

[141] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin, "Placing search in context: The concept revisited," *ACM Transactions on information systems*, vol. 20, no. 1, pp. 116–131, 2002.

[142] G. Halawi, G. Dror, E. Gabrilovich, and Y. Koren, "Large-scale learning of word relatedness with constraints," in *In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 1406–1414.

[143] T. Pedersen, S. Patwardhan, and J. Michelizzi, "Wordnet::similarity - measuring the relatedness of concepts," in *In Proceedings of the National Conference on Artificial Intelligence*, vol. 4, 2004, pp. 1024–1025.

[144] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[145] E. Agirre, O. L. De Lacalle, C. Fellbaum, S.-K. Hsieh, M. Tesconi, M. Monachini, P. Vossen, and R. Segers, "Semeval-2010 task 17: All-words word sense disambiguation on a specific domain," in *In Proceedings of the 5th international workshop on semantic evaluation*, 2010, pp. 75–80.

[146] P. Edmonds and S. Cotton, "Senseval-2: overview," in *In Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, 2001, pp. 1–5.

[147] B. Snyder and M. Palmer, "The english all-words task," in *In Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 2004, pp. 41–43.

[148] S. Pradhan, E. Loper, D. Dligach, and M. Palmer, "Semeval-2007 task-17: English lexical sample, srl and all words," in *In Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007)*, 2007, pp. 87–92.

[149] R. Navigli, D. Jurgens, and D. Vannella, "Semeval-2013 task 12: Multilingual word sense disambiguation," in *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, 2013, pp. 222–231.

[150] A. Moro and R. Navigli, "Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking," in *In Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, 2015, pp. 288–297.