A SEMANTIC SENTENCE SIMILARITY ESTIMATION APPROACH FOR THE BIOMEDICAL DOMAIN

by

Gizem Soğancıoğlu

B.S., Computer Engineering, Hacettepe University, 2013

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2016

A SEMANTIC SENTENCE SIMILARITY ESTIMATION APPROACH FOR THE BIOMEDICAL DOMAIN

APPROVED BY:

Assist. Prof. Arzucan Özgür .....
(Thesis Supervisor)

Assist. Prof. Cüneyd Tantuğ .....

Dr. Suzan Üsküdarlı .....

DATE OF APPROVAL: 06.06.2016

to my twin

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Arzucan Özgür who has been much more than an adviser to me during this study. She patiently devoted her valuable time to me and introduced me how to make research in an enjoyable way. Whenever I got desperate, she taught me to look on the bright side of things with her endless motivation and pure heartiness in her smile. Thanks to her, research always will be in my life in the future.

I am very grateful to my thesis committee members; Suzan Üsküdarlı and Cüneyd Tantuğ for their valuable recommendations. Suzan Üsküdarlı showed me how anything can be fabulous if you do it with your whole passion. Besides research world, she taught me that seeing anything with a different point of view is much more enjoyable in the real world. Due to her, I have learned to reply the question "How is it going?" with an answer "Let's eat chocolate".

I am very thankful to all my colleagues; Sacide, Erdem, Onuralp, Gülsün, Seçil, Mehmet in R&D at Yapı Kredi Technology, for listening to me patiently during the final stages of this study. My special thanks go to my teammate Bilge Köroğlu for her warm friendship and support and to my boss Onur Ağın for his undeniable support.

I have heartfelt thanks for my biggest supporter Hakime Öztürk who reflected her beautiful energy filled with goodness on me. Even in the times she was very busy with her stuff, she never gave up being a part of this study and believing in it. More importantly, she always motivated me in each stage of this study and has been a valuable friend to me.

I would like to thank Onur Yanar who was almost always with me and believed in me. I want to express my gratitude to my best friends: Funda Yıldırım and Simge Alkut for being an important part of my life.

The most cheerful thanks go to the most enjoyable man ever, Jesus Lago Garcia. He has even written a song with the lyrics "you are the best, alayına rest gizem" to make me happy.

I would like to thank a very special woman, Kübra Eren, for spreading colorful butterflies to my master memories.

Thank you is never enough to explain how grateful I am to my family for their life-long support. I thank my grandparents for filling my life with full of compassion. Due to them, I remember my childhood with a huge smile on my face. I am very thankful to my mother for having the warmest arms and heart and being more than a great mom to us! Life is so easy when she is with us. I thank my aunt for always making me feel her warm love. I also thank Süleyman Baş and my uncle for their support.

I am more than thankful to my one and only cousin, Fulden Özaras, who has unbelievably supported me. To provide me focus only on my thesis, she has done lots of house works at my home. She played the most enjoyable and depressive songs at the same time and made my last days unforgettable. In particular, without her help during the last days, writing this thesis would have not been possible.

I am very grateful to my faithful dude, Çino, for never leaving me alone. He never stopped loving me even in the times I was a little bit crazy and selfish during this study.

I dedicate this thesis to my twin sister who loves me as I am. She is shining my world with her little diary. What a beautiful thing to have you, my twin! Nothing is impossible when you hold my hand. My all success and power belong to you.

TUBİTAK-BİDEB 2210 scholarship program is gratefully acknowledged.

ABSTRACT

A SEMANTIC SENTENCE SIMILARITY ESTIMATION APPROACH FOR THE BIOMEDICAL DOMAIN

During the last decades, the use of semantic text similarity has been adopted as a major component in many Natural Language Processing tasks, including text retrieval, summarization, and document categorization. Integration of semantic information acts as a powerful tool for a better understanding and structuring of text. Among the many domains that benefit from text mining studies, biomedical literature is one of the most challenging areas because of its domain-specific language. As an inevitable result of the complex nature of the biomedical literature, domain-specific adaptations are crucial requirements. There are several semantic text similarity approaches that have been applied on the word-level. However, and to the best of our knowledge, there has not been any research on sentence-level semantic similarity in the biomedical domain. Furthermore, our experimental results revealed that domain-independent state-of-the-art approaches in sentence-level semantic similarity do not effectively cover biomedical knowledge and produce poor results. In this study, we propose several different approaches for domain-specific semantic sentence-level similarity computation, including measures utilizing distributional vector representations of sentences, methods combining general and domain specific ontologies, as well as a supervised approach exploiting high-level features. Our proposed methods are evaluated using a manually annotated data set which consists of 100 sentence pairs from biomedical literature. The experiments showed that the supervised semantic similarity computation approach obtained the best performance and improved over the previous domain-independent systems up to 42.6% in terms of the Pearson correlation metric.

ÖZET

BİYOMEDİKAL ALANDA ANLAMSAL CÜMLE BENZERLİĞİ HESAPLAMA YÖNTEMİ

Son yıllarda, anlamsal benzerlik yöntemlerinden, metin getirimi, otomatik özetleme, belge sınıflandırma gibi doğal dil işlemesi problemlerinin bir çok alanının önemli bir parçası olarak yararlanılmaktadır. Anlamsal bilginin katılması, metnin anlaşılması ve yapılandırılması için güçlü bir araçtır. Metin madenciliğinden yararlanan çalışma alanları arasında, biyomedikal literatürü kendine özgü dilinden dolayı en zorlu alanlardan birisidir. Biyomedikal literatürün karmaşık doğasının sonucu olarak, alana özgü uyarlamalara olan gereksinim kaçınılmazdır. Biyomedikal alan kapsamında, bu alana özgü bir çok kelimeler arası anlamsal metin benzerliği yöntemi bulunmaktadır. Ancak, bilgimiz çerçevesinde, literatürde biyomedikal alana özgü geliştirilmiş cümleler arası anlamsal benzerlik hesaplama yöntemi bulunmamaktadır. Bunun yanı sıra, yapmış olduğumuz deneyler, alandan bağımsız olarak geliştirilmiş en son çalışmaların başarısız sonuçlar ürettiğini göstermektedir. Çalışmamızda, biyomedikal alana özgü cümleler arası anlamsal benzerlik ölçümü için dağılımsal cümle vektörlerine dayanan bir yaklaşım, genel ve alana özgü ontolojileri kullanan bir yöntem ve üst düzey öznitelikler ile eğitilmiş güdümlü makine öğrenmesi tabanlı bir yaklaşım önerilmektedir. Önerilen yöntemler biyomedikal alandan 100 tane cümle ikilisinden oluşan elle etiketlenmiş veri kümesi üzerinde değerlendirilmiştir. Deney sonuçları, önermiş olduğumuz güdümlü anlamsal benzerlik hesaplayıcı yöntemimizin, alandan bağımsız sistemlere kıyasla en yüksek başarıyı elde ettiğini ve Pearson Korelasyon metriğine göre %42.6 başarıyı arttırdığını göstermektedir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iv
ABSTRACT vi
ÖZET vii
LIST OF FIGURES xi
LIST OF TABLES xiii
LIST OF ACRONYMS/ABBREVIATIONS xiv
1. INTRODUCTION 1
1.1. Motivations of the study 2
1.2. Contributions of the study 4
1.3. Overview of the study 5
2. BACKGROUND 6
2.1. String Similarity Methods 6
2.1.1. Term-based Similarity Measures 6
2.1.2. Character-based Similarity Measures 7
2.2. Ontology-based Word-level Measures 8
2.2.1. Path-based Measures 8
2.2.2. Path&Depth based Measures 10
2.2.3. Information Content based Measures 11
2.2.4. Definition based Measures 12
2.3. WordNet and UMLS Ontologies 13
2.3.1. WordNet 13
2.3.2. Unified Medical Language System (UMLS) 14
2.4. Learning Models 16
2.4.1. Multi Layer Perceptron 16
2.4.2. Random Forest 17
2.4.3. Linear Regression 18
2.4.4. Support Vector Machine 19
3. RELATED WORK 20
3.1. Word-level Semantic Similarity 20

3.2. Sentence-level Semantic Similarity 22
4. DATA PREPARATION 27
5. METHODOLOGY 31
5.1. Preprocessing 31
5.2. String Similarity Methods 32
5.3. Distributional Vector Models 33
5.3.1. Latent Semantic Analysis 33
5.3.2. Paragraph Vector 34
5.4. Ontology-based Similarity Methods 36
5.4.1. WordNet-based Similarity Module 36
5.4.1.1. WordNet-based Word-level Similarity 37
5.4.2. UMLS-based Similarity Module 39
5.4.2.1. UMLS-based Word-level Similarity 39
5.4.3. Algorithm for the Adaptation of Word Similarities to Sentence-level 42
5.4.3.1. A Walk-Through Example 42
5.4.4. Combined Ontology Methods 43
5.4.4.1. Word-level combined ontology method 44
5.4.4.2. Sentence-level combined ontology method 44
5.5. Supervised Approaches 45
5.5.1. Features 45
5.5.1.1. Low-level Features 45
5.5.1.2. High-level Features 47
6. EXPERIMENTS AND RESULTS 49
6.1. Evaluation Metric 49
6.2. Results 50
6.2.1. String Similarity Measures 51
6.2.2. Distributional Semantic Vector Models 52
6.2.3. Ontology-based Similarity Systems 54
6.2.4. Supervised Approaches 57
7. CONCLUSION AND FUTURE WORK 61
7.1. Conclusions 61

7.2. Future Studies 63
REFERENCES 65
APPENDIX A: SYSTEM RESULTS FOR A SUBSET OF THE DATA SET . 76

LIST OF FIGURES

Figure 2.1. Hierarchical relationships among a small subset of proteins and antibiotics 9

Figure 2.2. UMLS ONTOLOGY&RESOURCES 14

Figure 2.3. Multi Layer Perceptron 17

Figure 4.1. Collected Data Set Format 28

Figure 4.2. Distribution of the Similarity Scores in the Data Set 29

Figure 5.1. String Similarity Measures 32

Figure 5.2. Command for Training the LSA Vectors 34

Figure 5.3. Command for Training the Paragraph Vectors 35

Figure 5.4. WordNet-based Similarity Method 37

Figure 5.5. Pseudocode for Computing Similarity between Word Pairs in WordNet 38

Figure 5.6. UMLS-based Similarity Method 39

Figure 5.7. Command for Running UMLS:Similarity Web Interface 42

Figure 5.8. Algorithm for Building a Semantic Vector of a Sentence 43

Figure 5.9. Illustration of the proposed algorithm which constructs semantic vectors of sentences 44

Figure 5.10. Sentence-level Combined Ontology Method 45

Figure 5.11. Sentence-level Combined Ontology Method Pseudocode 46

Figure 5.12. Supervised Similarity Method exploiting low-level features 47

Figure 5.13. Pseudocode of the High-level Supervised Approach 48

Figure 5.14. Supervised Similarity Method Exploiting High-level Features . . . 48

Figure 6.1. Correlation Between Systems and Ground Truth 60

LIST OF TABLES

Table 3.1. Available Measures through the SEMILAR Online Demo 26

Table 4.1. Annotation Guideline 28

Table 4.2. Example Annotations 30

Table 5.1. The available sources and relations via the UMLS:Similarity web interface 40

Table 5.2. Word-level UMLS Similarity 41

Table 6.1. Correlation scores among annotators 50

Table 6.2. Correlation scores for domain-independent state-of-the-art systems 51

Table 6.3. Results of the String Similarity Measures 53

Table 6.4. Correlation Scores for Paragraph Vector and LSA 54

Table 6.5. Correlation scores for WordNet based similarity systems 55

Table 6.6. Correlation scores for UMLS based similarity systems 56

Table 6.7. Correlation scores for combined ontology method 57

Table 6.8. Correlation scores for supervised similarity system 59

Table A.1. Ground truth and supervised system scores for selected pairs . . . 76

LIST OF ACRONYMS/ABBREVIATIONS

BioNLP	Biomedical Natural Language Processing
CUIs	Concept Unique Identifiers
GO	Gene Ontology
GT	Ground Truth
JCN	Jiang and Conrath
LCH	Leacock Chodorow
LCS	Longest Common Substring
LR	Linear Regression
LSA	Latent Semantic Analysis
MeSH	Medical Subject Headings
MLP	Multi Layer Perceptron
NLP	Natural Language Processing
OAS	Open Access Subset
OMIM	Online Mendelian Inheritance in Man
POS	Part-Of-Speech
RF	Random Forest
SVM	Support Vector Machine
SVD	Singular Value Decomposition
SVR	Support Vector Regression
UMLS	Unified Medical Language System
WP	Wu and Palmer

1. INTRODUCTION

Semantic text similarity is a research problem which aims to calculate similarities among texts based on the similarities of their meanings and semantic content, rather than their shallow or syntactic representation. Metrics on semantic text similarity have undertaken a crucial role in many Natural Language Processing (NLP) applications. Three sample applications are described below.

- In recent years, community-based question and answer services on the Web where people answer other peoples' questions have become very popular. By using a huge amount of these available data, a method for finding questions in the archive that are semantically similar to a given question is proposed in [1]. In this study, a semantic similarity approach helps finding high quality answers from the archive as a reply to a user's question. In other words, question answering is performed with the help of a semantic similarity method.
- The task of machine translation evaluation is highly related to the sentence-level semantic similarity problem. Many semantic sentence similarity approaches have been proposed for machine translation evaluation. In [2], machine translation evaluation is directly performed by using a semantic sentence-level similarity method. First, a similarity score between items such as words or n-grams across two sentences is calculated. Then, maximum weight matching algorithm is applied to match each item in one sentence with at most one item in another sentence enabling the maximum overall similarity score. In addition to studies that use semantic text similarity for machine translation evaluation, there are studies that propose using well-known machine translation evaluation measures for the calculation of semantic similarity among sentences [3].
- Sentence similarity computation is one of the key problems in multi-document summarization, which aims to select the most informative sentences that reflect the main characteristics of documents. Many approaches exploiting sentence-level similarities to obtain the most informative sentences have been proposed. For example, sentence level semantic similarities have been used to cluster sentences

for multi-document summarization in [4]. Then, the most informative sentences in each group have been selected to form a summary [4].

These exemplary applications show that sentence similarity computation is an important component in many NLP tasks. Consequently, it has attracted the attention of researchers in the recent decades. Various measures on different text levels have been studied so far. Thanks to the ontologies providing semantic relations between concepts, a number of ontology based similarity measures have been developed. Besides these metrics, approaches using large corpora to learn the semantic relations between text fragments have also been proposed.

1.1. Motivations of the study

In the last decades, finding and reading relevant information in biomedical literature requires painstaking work due to the extremely rapid publication rate. Since publications and clinical records are mostly written in text, NLP became crucial in biomedical research to solve the information explosion problem, as it can efficiently convert unstructured text into structured knowledge. This situation has led researchers to a new area named Biomedical NLP (BioNLP) which aims to develop NLP methods for various kinds of biomedical applications. Various research fields of NLP such as text retrieval, automatic summarization, named entity recognition, question answering have been studied specifically for the biomedical domain.

Similar to studies in NLP, BioNLP researches also require semantic similarity measures as a core component of its many applications. As text processing studies in biomedical literature require new approaches specifically developed for their domain, semantic text similarity measures to be used in BioNLP studies call for domain-specific approaches, since domain-independent measures exploit domain-independent ontologies or corpora which do not cover biomedical concepts, whereas biomedical concepts comprise the most valuable part of sentences for semantic similarity estimation.

As an example, consider the following two sentences.

- S1: This form of necrosis, also termed necroptosis, requires the activity of receptor-interacting protein kinase 1 and its related kinase 3 [5].
- S2: Moreover, other reports have also shown that necroptosis could be induced via modulating RIP1 and RIP3 [6].

The example sentences S1 and S2 relate to the same topic and are quite similar to each other. Protein kinase 1 in S1 is the same concept as RIP1 in S2, likewise 'kinase 3' and RIP3 refer to the same biomedical term. Despite of being the most indicative words for the semantic similarity score of the overall sentence, domain-independent measures can neither recognize these concepts nor give high weight to them. Another remarkable example is provided below.

- `T1:` miR-Vec constructs were described before, and Dnd1 open-reading frames were cloned as described into a pCS2-based CMV expression vector to contain a double carboxy-terminal HA tag [7].
- `T2:` The pMSCV-blast-miR plasmids, containing either hsa-miR-376a1 human miRNA or control miRNA (hTR-human telomerase RNA), were constructed as described previously [8].

From a biomedical perspective, sentences T1 and T2 are not equivalent and they relate to different things. On the other hand, domain-independent measures find common words such as 'construct, describe, contain' and find similar words such as 'previously, before' in the two sentences. As a consequence, domain-independent measures find the sentence pair semantically highly similar in spite of the fact that these words do not determine the overall meaning in the sentence.

Since in the biomedical literature, sentences consist of both biomedical and domain independent words, it is clear that new approaches are required. There are some researches on semantic word similarity adapted for biomedical domain. Nevertheless and to the best of our knowledge, there has not been any study on semantic sentence

level similarity for the biomedical domain. As a result, the biggest motivation in this study was to develop a semantic sentence similarity measure for the biomedical domain, which may play in turn a critical role for various tasks of BioNLP. In particular, we propose new approaches specifically adapted for the biomedical domain which can be categorized into three areas: combined ontology based measures, distributional vector models, and supervised systems. Moreover, since there is no a test collection for the evaluation of semantic sentence similarity measures in the biomedical domain, in this work, we crafted our own test data set as mentioned in Chapter 4 consisting of 100 sentence pairs. We believe that algorithms can objectively be compared with each other by measuring their performance on this data set. Furthermore, this test collection, which we aim to share with other researchers, will promote and support research in this area.

Besides the increasing researches in various fields of BioNLP, a number of challenge evaluations, which enable state-of-the-art studies to improve, have been organized. A recent study [9] reviews different challenges held from 2002 to 2014 on (BioNLP) researches. Researchers expect to see even new challenges which address different user needs in biomedical research. By this study, we believe that a new research field is proposed and there is an open room for improvements that may be provided by further studies in this area.

1.2. Contributions of the study

In this thesis, we

- investigate the usability of domain-independent state-of-the-art semantic similarity systems for biomedical literature and show that they produce poor results.
- adapt domain-independent semantic similarity computation approaches to biomedical domain as a first study.
- evaluate distributional vector models, ontology-based approaches, and string similarity measures for semantic similarity computation.
- demonstrate that hybrid approaches combining several measures outperform sin-

gle ones because of being able to utilize each measure and show that a supervised machine learning based approach performed best among others.

- compare popular ontology-based measures which produce semantic similarity scores between words for the sentence-level semantic similarity computation task.
- compare different regression learning models for the supervised system.
- evaluate well-known string similarity measures which take into consideration lexical rather than semantic similarity.
- provide a test collection consisting of 100 sentence pairs from biomedical literature. The test collection is constructed manually and annotated by 5 different human experts. It is available for other researchers as the first data set prepared for semantic sentence similarity studies in the biomedical domain.

1.3. Overview of the study

The rest of this study is organized as follows. In the next chapter, we review the methods employed in our systems. In Chapter 3, a literature review on semantic word and sentence similarity is provided. In Chapter 4, we present the characteristics of the test collection crafted by us and the annotation guideline which were given to the human experts. In Chapter 5, we present our approaches on semantic sentence similarity, namely combined ontology-based semantic similarity and supervised system exploiting high-level features. Moreover, we also briefly explain the methods that we evaluate on our data set including string similarity metrics and distributional models. In Chapter 6, we provide experimental results for each measure and discuss the research findings. Finally, we conclude the study with a summary of findings and future research pointers.

2. BACKGROUND

In this chapter, we will give some background information for the better understanding of our approaches introduced in Chapter 5. Firstly, we will present the well-known string similarity measures. Then, we will describe the popular ontology-based word-level similarity algorithms in Section 2.2. Third, we will introduce the key aspects of WordNet and UMLS ontologies including the semantic relations encoded between concepts and sources in Section 2.3. Finally, we will review the theory of learning regression models, which are exploited for our supervised systems, in Section 2.4.

2.1. String Similarity Methods

String similarity metrics measure the similarity or distance between two text strings for comparison or approximate string matching. Various string similarity measures based on characters or terms have been evaluated in this study and expressed in Section 5.2. String similarity measures can be categorized into two approaches: term-based and character-based. In this section, we will present different character-based and term-based string similarity methods.

2.1.1. Term-based Similarity Measures

Cosine Similarity [10] is a measure of similarity that calculates the cosine of the angle between two vectors in vector space (Equation 2.1).

$$similarity = \cos\theta = \frac{\overrightarrow{X} \cdot \overrightarrow{Y}}{||\overrightarrow{X}||||\overrightarrow{Y}||} \tag{2.1}$$

Jaccard Similarity [10] measures the similarity between two sets and is computed as the number of common terms over the number of all unique terms in both sets (Equation

2.2).

$$similarity = JAC(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.2}$$

Overlap Coefficient [11] is a similarity measure that differs from Jaccard Similarity with being divided by the size of the smaller sized of the two sets (Equation 2.3).

$$similarity = Overlap(A, B) = \frac{|A \cap B|}{|Min(|A|, |B|)|} \tag{2.3}$$

Dice's Coefficient [12] is computed as twice the number of shared terms divided by the total number of terms in both sets (Equation 2.4).

$$similarity = Dice(A, B) = \frac{2|A \cap B|}{|A| + |B|} \tag{2.4}$$

Block Distance [13], also known as Manhattan Distance, computes the distance between two points by summing the differences of their corresponding components. The Equation for block distance between a point a=$(a_1, a_2, ..., a_n)$ and b=$(b_1, b_2, ..., b_n)$ in n-dimensional space is:

$$BD(a, b) = \sum_{i=1}^{n} |a_i - b_i| \tag{2.5}$$

2.1.2. Character-based Similarity Measures

QGram Distance [14] is typically used in approximate string matching by "sliding" a window of length q over the characters of a string to create a number of 'q' length grams for matching. A match is then rated as the number of q-gram matches within the second string over possible q-grams.

Levenshtein Distance [15] is a simple edit distance which consists of the operations for transforming one of the given strings to another, where an operation is defined as

an insertion, deletion, substitution, or copying of a character. The distance is defined as the minimum number of required operations to change one string into another.

The Smith-Waterman algorithm [16] is used mostly in bioinformatics problems since it performs local sequence alignment for detecting the similar regions between two strings.

The Longest Common Subsequence (LCS) algorithm [17] finds the common subsequence with the maximum possible length of two strings. The algorithm does not require the characters in the common subsequence to be consecutive.

The Needleman-Wunch algorithm [18] is similar to Levenshtein distance but differs by adding additional gap cost for insertion and deletion operations. So, Levenshtein distance can be seen as Needleman-Wunch if the cost of gap equals 1.

2.2. Ontology-based Word-level Measures

Ontology based similarity measures can be categorized into four categories namely path-based, path&depth based, information content based, and definition based approaches. Both path and path&depth based approaches utilize the structure of the taxonomy, whereas information content based approaches use extra information learned from the corpus statistics. The following briefly describes the ontology-based similarity metrics that are employed in our proposed method (Section 5.4).

2.2.1. Path-based Measures

The Path [19] algorithm measures the semantic similarity of two concepts by calculating the shortest path between them in a taxonomy. The intuition behind the algorithm is that the shorter the path between concepts in a hierarchy the more similar they are.

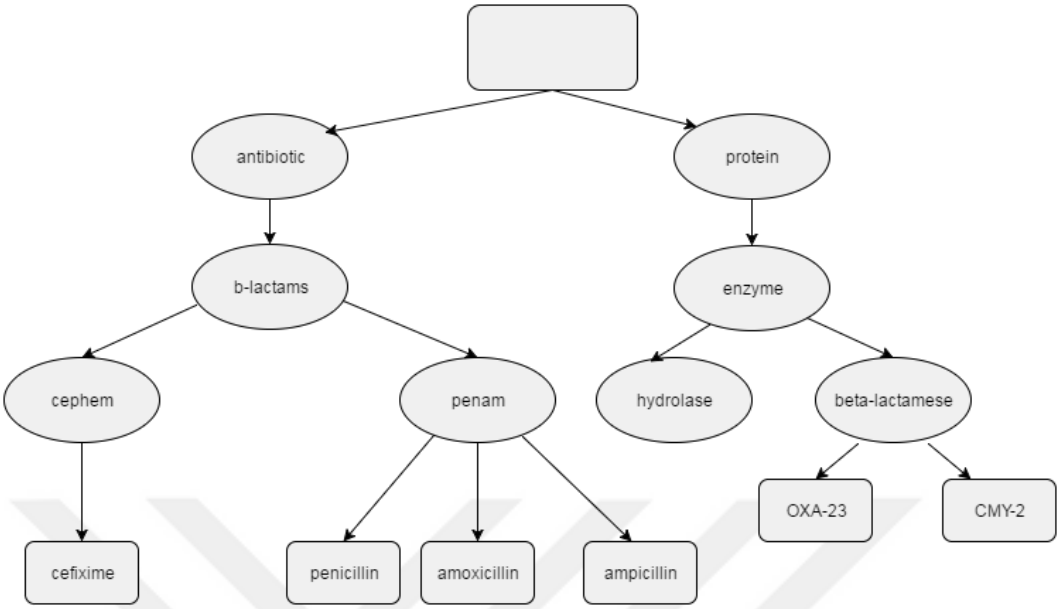


Figure 2.1. Hierarchical relationships among a small subset of proteins and antibiotics

$$Sim_{Path}(c_1, c_2) = (2 * depth_{max}) - len(c_1, c_2) \tag{2.6}$$

In the Equation 2.6, $depth_{max}$ refers to the maximum depth of the taxonomy and the len function computes the shortest path from concept c_1 to c_2. The semantic distance between the terms “protein" and “b-lactams" (Figure 2.1) is computed as:

$$Sim_{Path}(protein, \text{b-lactams}) = (2 * 5) - 4 = 6 \tag{2.7}$$

The shortest path between c_1 and c_2 counts all nodes between them - including themselves. Since the maximum depth of the taxonomy is constant, this measure does not take into consideration the specificity of concepts. According to the definition, $len(c_1, c_2)$ is equal to 4 and $depth_{max}$ is 5.

Conceptual Distance (Cdist) [20] computes the number of edges between two concepts in the UMLS ontology. This approach uses the definition of concept parent-child relations for measuring similarity.

2.2.2. Path&Depth based Measures

Path&Depth based measures differ from path-based approaches by using the path along with depth knowledge which is also learned from the structure of the taxonomy. Unlike the path-based measures, these measures account for specificity due to the depth feature.

Leacock & Chodorow (LCH) [21] similarity measure is defined as:

$$Sim_{LCH}(c_1, c_2) = -\log \frac{len(c_1, c_2)}{2 * D} \tag{2.8}$$

In Equation 2.8, D represents the maximum depth of the taxonomy, while len returns the length of the shortest path between the two concepts. Depth of a concept is calculated by counting the nodes from the root to the concept.

Wu and Palmer (WP) [22] similarity between concepts c_1 and c_2 is measured as twice the depth of the lowest common subsumer (LCS) of the given concepts divided by the sum of depth of c_1 and c_2.

$$Sim_{WP}(c_1, c_2) = \frac{2 * depth(LCS(c_1, c_2))}{depth(c_1) + depth(c_2)} \tag{2.9}$$

For understanding the idea behind the algorithms using the depth feature, the following example which uses WP metric to illustrate the effect of depth is useful.

$$Sim_{WP}(cephem, ampicillin) = (2 * 3)/(4 + 5) = 0.66 \tag{2.10}$$

$$Sim_{WP}(antibiotic, enzyme) = (2 * 1)/(2 + 3) = 0.40 \tag{2.11}$$

$$Sim_{Path}(cephem, ampicillin) = 10 - 4 = 6 \tag{2.12}$$

$$Sim_{Path}(antibiotic, enzyme) = 10 - 4 = 6 \tag{2.13}$$

While the Path algorithm gives the same semantic similarity score for the two pairs which have different specificity, WP estimates that cephem and ampicillin are more similar than antibiotic and enzyme. The result of WP metric seems reasonable for this example, since the path between deeper concepts causes less semantic distance.

2.2.3. Information Content based Measures

The depth feature gives an idea about the specificity of a concept. However, the frequency of a concept in a thesaurus also determines the specificity of the concept. With the motivation of this idea, information content is used for measuring the semantic similarity between concepts. Information content (IC) of a concept c is defined as the negative of the logarithm of the probability of encountering the concept in a corpus.

$$IC(c) = -\log(p(c)) \tag{2.14}$$

The probability of encountering concept c is given,

$$p(c) = freq(c)/N \tag{2.15}$$

In Equation 2.15, N denotes the total number of words, while freq(c) is the number of occurrences of concept c in a taxonomy.

Resnik [23] similarity measure is determined as the information content of the lowest common subsumer of concepts, c_1 and c_2.

$$Sim_{Resnik}(c_1, c_2) = IC(LCS(c_1, c_2)) \tag{2.16}$$

Lin [24] similarity between concepts, c_1 and c_2, is calculated as twice the information content of the lowest common subsumer of the concepts over the sum of the

information content of c_1 and c_2.

$$Sim_{LIN}(c_1, c_2) = \frac{2 * IC(LCS(c_1, c_2))}{IC(c_1) + IC(c_2)} \tag{2.17}$$

Jiang and Conrath (JCN) [25] measures the semantic similarity between concepts, c_1 and c_2, using the Equation below which uses the information content of concepts and their lowest common subsumer.

$$Sim_{JCN}(c_1, c_2) = \frac{1}{IC(c_1) + IC(c_2) - 2 * IC(LCS(c_1, c_2))} \tag{2.18}$$

2.2.4. Definition based Measures

Definition based methods are mostly used for measuring the semantic relatedness between two concepts rather than similarity. Since definition based measures utilize the glossary of concepts, even if there is not an is-a relation between concepts, this methods may find them highly related.

The Lesk [26] algorithm is proposed as a solution for word sense disambiguation. According to the Lesk algorithm, semantic similarity of two concepts is measured with the overlap between the corresponding dictionary definitions. Concepts from different ontologies can be measured by the Lesk algorithm due to the fact that it is dependent on the concept definition rather than taxonomy.

The Vector [27] algorithm creates a co-occurrence matrix for each word used in the ontology glosses from a given corpus, and then represents each gloss/concept with a vector that is the average of these co-occurrence vectors.

2.3. WordNet and UMLS Ontologies

In the experiments referred in Section 5.4, WordNet and UMLS ontologies were exploited to calculate the similarity scores between concepts. In this section, to be able to understand the overall structure and types of relationships in the WordNet and UMLS ontologies, we give some basic information about them in the following Sections 2.3.1 and 2.3.2.

2.3.1. WordNet

WordNet [28] is a large English lexical thesaurus that has been widely used in measuring semantic similarity by using the measures mentioned in Section 2.2. According to the structure of WordNet, each word consists of a form 'f' which is a string and a sense 's' represented by a set of synonyms that have that meaning. Words in WordNet are categorized according to syntactic properties such as verb, noun, adjective, and adverb. Since the same words can be interpreted as different part-of-speech (POS) tags according to the contexts they occur in, this syntactic categorization allows to save same word with each possible POS tags separately in a taxonomy. Moreover, words and word senses are connected to each other with various types of relationships. The most used ones for measuring semantic similarity are listed below:

- Synonymy is a basic relation since set of synonyms are used to represent word senses.
- Hyponymy and Hypernymy represents the relations between the word and it's super-name or sub-name.
- Antonymy relations show the connection between the name and it's opposite-name.

By using the rich vocabulary and encoded relations between words, all measures referred to n Section 2.2 are implemented for WordNet Ontology. There are various software packages that implement these measures such as:

- WS4J [29]
- WordNet:Similarity [30]
- JWNL [31]

2.3.2. Unified Medical Language System (UMLS)

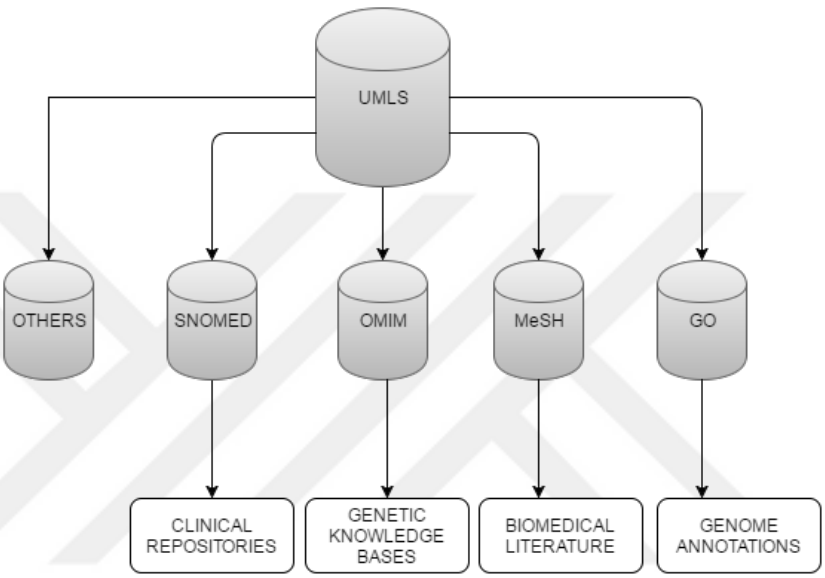


Figure 2.2. UMLS ONTOLOGY&RESOURCES

UMLS [32] is a comprehensive thesaurus consisting of more than 1.7 million biomedical concepts. As shown in Figure 2.2, it comprises of the vocabulary sources on specialized topics such as MeSH consisting of medical subject headings, OMIM containing genetic knowledge bases or SnomedCT which consists of the concepts belonging to clinical repositories. Since UMLS consists of various terminology sources, some concepts can overlap. In other words, the same concept can be belonging to different sources. To be able to use multiple sources as a single resource in Metathesaurus, Concept Unique Identifiers (CUIs) are assigned to the concepts.

UMLS has three tools which are named as Metathesaurus, Semantic Network, and Specialist Lexicon. These tools are explained below briefly.

Metathesaurus is a large biomedical thesaurus consisting of inter-related concepts and the biggest component of UMLS. Metathesaurus is categorized by the concepts and

their semantic types such as meaning. Synonymous terms for the same concept are linked to each other and clustered together. Moreover, various types of relationships between different concepts are defined in the thesaurus. Relationships between concepts can be categorized into 3 types which are hierarchical such as 'is-a', associative as 'caused by', and statistical. These concepts come from one or more of the source vocabularies such as OMIM, SnomedCT or MeSH. Therefore, if a concept does not occur in a source vocabulary, it is also not defined in the Metathesaurus.

Semantic Network: Semantic relations and a set of broad categories encoded in Metathesaurus are provided by the Semantic Network knowledge source of UMLS. While information about a specific concept is provided in Metathesaurus, the Semantic Network provides information about the basic semantic categories, which may be assigned to these concepts, and the set of relationships that may hold between the semantic types.

Specialist Lexicon has been developed with the intent of being used by NLP systems. It provides an English lexicon containing many biomedical terms alongside commonly used general words with the syntactic and morphological information for each word.

Most of the measures referred in Section 2.2 are originally based on WordNet ontology. Since WordNet is a domain-independent resource, it does not cover most of the biomedical terms. To make measures more effective in domain-specific tasks, they have been applied to biomedical ontologies. There are various tools and software packages available on the web. Some of the tools implement the measures using specific source vocabularies such as GO or DO while some use all sources in UMLS ontology. Some of the libraries and tools that implement semantic similarity measures in biomedical ontologies are listed below.

- FastSemSim (Gene Ontology) [33]
- GoSim (Gene Ontology) [34]
- DoSim (Disease Ontology) [35]

- UMLS:Similarity (UMLS) [36]

2.4. Learning Models

In this section, we present the learning models that we have used for our supervised learning system the details of which are presented in Section 5.5. As learning models, Multi Layer Perceptron (MLP), Linear Regression (LR), Support Vector Machine (SVM) and Random Forest (RF) algorithms are used and the related background information is given below.

2.4.1. Multi Layer Perceptron

Multi Layer Perceptron (MLP) [37] is a feed forward network model that comprises multiple layers of nodes. MLP has the ability to distinguish data that are not linearly separable which makes them powerful. It consists of input and output layer and desired number of hidden layers depending on the model. The layers are connected to each other from input to output layer and all connections are weighted with a real number. The nodes of the network are neurons with an activation function.

Figure 2.3 illustrates a MLP. The variables x_1, x_2 to x_n represent input neurons that are no target of any connections. Y represents the output neuron of the network that is no source of any connections. Arrows in the network represent the connections between the layers and all of them are weighted. Small blue circles are neurons in the network with an activation function.

In an intuitive way, MLP tries to find a function which can map input to output data. This process is called training. To do so, it uses Backpropagation learning technique with gradient descent or its variations [38]. Due to calculation of gradients, activation functions of neurons should be differentiable and continuous.

The training phase consists of two passes namely, a forward and a backward pass. In the forward pass, input is fed to the network and the output of the network is

calculated within the current weights. Then the error which is the difference between real and predicted output is calculated. In the backward pass, by backpropagation of the error towards input layer, the gradient of the error with respect to the weights are calculated. It gives us how much each weight effects the error. Then weights are updated with gradient descent method and this two phases are done iteratively until the error is small enough.

MLP performs very well in many natural language processing tasks and is thus widely used.

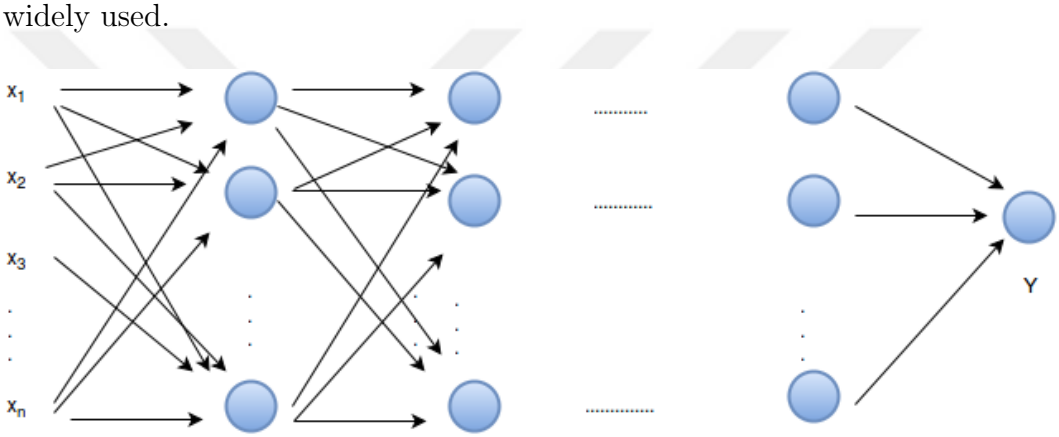


Figure 2.3. Multi Layer Perceptron

2.4.2. Random Forest

Random Forest [39] is a widely used learning method for classification and regression tasks. It is based on the idea that the combination of learning models gives better accuracy which is called ensemble learning methods. Random Forest uses many different decision trees that are trained on random subset of the training data and does classification and regression by combining the results of many decision trees.

Decision Trees are a powerful learning method that are widely used in many natural language processing tasks but lack generalization ability. A decision tree tends to overfit on the training data set. That is the reason why Random Forest comes into play. Combination of many decision trees aims to reduce the variance [40].

The algorithm of Random Forest is developed by Leo Breiman and Adele Cutler [41] and can be listed as in the following.

- Create random subsets of the training data.
- For each subset of the training data, construct decision trees.
- For test phase, test the data on the trained decision trees.
- Combine the result (majority vote for classification, averaging for regression) of decision trees to do inference.

2.4.3. Linear Regression

Linear regression is a very simple supervised learning algorithm which learns a linear function that fits the data best [42]. It learns a hypothesis which is a linear function that maps an input to the output that is Equationted as in the following for one feature problems:

$$h_z(x) = z_0 + z_1 x \tag{2.19}$$

In Equation 2.19, x is an input and z_0 and z_1 are the parameters that we need to learn. That's why learning a hypothesis corresponds to learning the parameters z_0 and z_1. To do so, linear regression defines the cost function as in the following:

$$\sum_{i=1}^{m} (h_z(x^{(i)}) - y^{(i)})^2 \tag{2.20}$$

As seen in Equation 2.20, the cost function is defined as the squared differences between the prediction of the hypothesis and the real output data. In an intuitive

way, we learn the hypothesis which maps an input to the output which is as close as possible to the real output value in the training data. Minimizing the cost function for the parameters z_0 and z_1 gives us the best hypothesis. To learn the parameters z_0 and z_1 that minimize the cost function, gradient descent method [43] is used.

2.4.4. Support Vector Machine

Support Vector Machine (SVM) [44] is a non-probabilistic learning model that is widely used in classification, regression, and clustering tasks. SVM constructs a hyperplane which separates the different classes in the training data. But this would not be sufficient for SVM since there are many or even sometimes infinite number of hyperplanes that can separate the different classes perfectly. SVM chooses the hyperplane which maximizes the margin and gives the smallest generalization error. Margin is defined as the perpendicular distance from the hyperplane to the closest training data points. To do so, SVM solves an optimization problem to find the best hyperplane which is called as decision boundary.

SVM can work with linearly separable data. To cope with non-linear data, SVM uses the kernel trick which transforms input data which are not linearly separable into a higher dimensional feature space. After transforming the input data, SVM applies linear separation there.

SVM can be used for regression tasks as well with slight modifications on the SVM classifiers [45]. It makes it possible by applying an alternative loss function which includes a distance measure.

3. RELATED WORK

The studies on semantic textual similarity can be categorized into three levels: word, sentence, and document. In this chapter, we will review the previous studies not only on the sentence-level, but also on the word-level since we will exploit the word-level similarities in our study. Previous studies on these areas are explained in detail in Sections 3.1 and 3.2. Since there is no previous research on semantic sentence-level similarity in the biomedical domain, we introduce sentence-level domain-independent measures in Section 3.2.

3.1. Word-level Semantic Similarity

Word-level similarity has been studied most among these three different levels. We can categorize the methods in this research area as corpus-based, ontology-based and hybrid approaches [46]. Corpus based approaches determine the similarity among words according to statistics learned from large corpora, while ontology-based approaches employ the information derived from semantic networks such as WordNet [28] or UMLS [32]. Unlike other methods, hybrid approaches exploit multiple similarity measures such as combining corpus-based and knowledge-based measures.

Among ontology-based approaches, the most popular semantic similarity methods are implemented and evaluated using WordNet. Wordnet is an English thesaurus that contains links between semantically related concepts. These semantic relations among concepts have a crucial importance since they can be utilized by many research areas such as text retrieval [47]. As a core component of many NLP studies, semantic word similarity using ontologies has attracted the researchers' greatest attention.

In 1989, Rada et al. [19] recommended an approach that was based on the path distance among concepts for semantic similarity on semantic networks. Due to some deficiencies of the path measure such as ignoring the specificity of the concepts, new approaches using the depth of the concepts in a graph were introduced [21,22]. Besides

this idea, measures as Resnik, Lin, Jiang using the information content [23–25], which are based on the frequency of the concept and also give information about the specificity of the concept, are considered to overcome the inefficiency of the previous similarity measures. However, all measures presented have their own drawbacks. In order to utilize the contributions of various measures, hybrid approaches and some adaptations to previous measures are proposed.

In the study [48], a new model based on a combined approach is presented. Their model considers both path length and IC value of concepts and the semantic similarity is calculated by summing the weighted scores of each part in the similarity Equation. Similarly, in another study [49], a similarity measure, which uses structural semantic information from WordNet and information content from a different source, is proposed. They showed that the proposed measures combining information sources, which are WordNet and Brown corpus, non-linearly outperformed the previous traditional measures.

As an adapted algorithm, an approach [50] that gives weights to the edges of WordNet ontology is suggested. The nearer edges to the root, which are weighted less, effect the overall similarity measure according to their Equation.

Besides the studies mentioned above on semantic similarity measures using ontology, detecting the relatedness of two concepts by using ontology is also considered and studied so far [26,27,51]. Since relations which are considered for relatedness measure are not restricted with the is-a relations, relatedness is a more general concept than similarity. In the studies [26,27], dictionary definitions are used to determine the similarity score. A review study on similarity measures in WordNet [52] discuss the advantages and disadvantages of measures using WordNet.

Due to the need for detecting the functional similarity between concepts rather than sequential similarity, semantic similarity measures for biomedical domain have been studied. Since WordNet does not effectively cover biomedical concepts, researchers needed to adapt the previous popular measures to biomedical ontologies such as GO

[53]. In the study [54], originally WordNet-based measures were adapted to SnomedCT which is the source vocabulary of UMLS. Path-based measures, IC based measures and vector measure are implemented and evaluated on their crafted small test set. The test set consists of 30 medical concept pairs and annotated by physicians and medical experts.

Besides single ontology based approaches exploiting either WordNet or UMLS ontology, calculation of the semantic similarity score across multiple ontologies has been studied. In the study [55], as a domain-independent ontology WordNet is used, while MeSH and SnomedCT are used as domain-specific biomedical ontologies. As a result of the experiments, it is observed that integrating information across multiple ontologies provided better coverage and outperformed the methods using a single ontology.

3.2. Sentence-level Semantic Similarity

Sentence-level methods can be categorized as:

(i) Those that consider word-level measures and calculate similarity between two texts by aggregating the similarities of word-pairs in texts
(ii) Those that view and model sentences as whole and calculate the similarity between these models.

Most of the previous systems are based on the first approach where word-level similarities are calculated, then some algorithms are applied to use these word-level scores to obtain the sentence-level score. In [56], several corpus based methods such as Latent Semantic Analysis (LSA) and knowledge based measures such as Resnik which are based on word-level similarities have been compared. Since there is no available data set to test the effectiveness of semantic text similarity metrics, the proposed measures have been evaluated on a paraphrase data set. By using similarity metrics, automatically identifying if two texts are paraphrases of each other is determined. The study demonstrated that semantic short-text similarity methods outperform methods based on simple lexical matching. Moreover, among semantic similarity metrics, the best

performing system has been obtained by combining several similarity metrics into one.

In another approach [57], a method which takes into account the semantic information, which is learned from knowledge sources such as WordNet and corpus statistics, and word-order information has been proposed. In this study, specifically domain-independent sentence-level similarities have been studied rather than short texts. Therefore, a small data set consisting of selected 30 sentence pairs was crafted to be able to measure the success of the proposed systems. Path algorithm which is proposed by Rada et al. [19] for the calculation of word-level similarities has been modified by combining it with depth feature in this study. Information contents of word pairs have been integrated to the Equation. Then, this calculated word-level similarities have been used for constructing sentence vectors for sentence pairs. We used a similar strategy for applying word-level similarities for sentence-level similarity. So, the details of the algorithm can be found in Section 5.9.

The SemEval Semantic Textual Similarity (STS) have organized task series [58–61] each year since 2012 for semantic textual similarity. A publicly available corpus of more than 14,000 sentence pairs have been developed over the four years. These sentence pairs have been labelled with a similarity score in a range [0-5] by human annotators. By this time, a total of 290 runs attending this shared task have been evaluated. Besides, many approaches, which have used the data set of SemEval organization and compared their systems with the best ranked runs in the task, have been proposed. Due to this task series and provided data set, the semantic sentence-level similarity problem, which has been studied less in comparison with word-level similarity, has recently attracted researchers' attention. Moreover, a huge amount of annotated data set enabled the applications of supervised approaches for semantic sentence-level similarity. Recent state-of-the-art methods on semantic sentence-level similarity often involve supervised approaches.

The best performing system [62] in terms of normalized Pearson correlation in SemEval 2012, have used support regression model with multiple features measuring word-overlap similarity and syntax similarity. As word-overlap measures, n-gram

overlap, WordNet-based word overlap, weighted word overlap weighted with the information content of words, vector space sentence similarity based on LSA, and greedy lemma alignment overlap have been used. Besides word-overlap similarities, syntactic features have been used due to the fact that words having the same syntactic roles in two sentences may contribute to the overall semantic similarity of the sentences. For extracting syntax similarity, a part-of-speech tagger has been used. Although focus of the study is word-overlap and syntactic features, some different features such as number overlap and named entity features have been added and the experiments showed that these added features have also increased the overall performance of the proposed system.

Among the submitted runs in SemEval 2013, the study [63] has obtained the best performance. The proposed measure is based on the boosting LSA similarity using WordNet according to predefined conditions. For aligning terms in two sentences, first, a POS tagger has been applied to obtain tags and lemmatize the sentences. The method using simple term alignment algorithm outperformed the support vector regression model combining multiple features.

In SemEval 2014, the best ranked system DLS@CU [64] was based on a previous study [65] proposing a monolingual word aligner. DLS@CU has presented an approach which uses the output of a word aligner for a sentence pair by taking the proportion of their aligned words as semantic similarity score. The idea behind using a word aligner is that semantic similarity is strongly related with the count of semantic components meaningfully aligned in sentences.

In SemEval 2015, a modified approach of DLS@CU, which was presented in SemEval 2014, ranked 1st again and it was implemented by the same research group. In this study, a supervised system using the output of the word aligner and compositional vector representation based on the Word2Vec [66] toolkit was presented. Combining the previous method with sentence vector similarity measure increased the performances of the previous study significantly.

Recently, in another approach [67], a supervised learning model using rich feature sets to predict the similarity between two sentences has been introduced. In this study, several unsupervised measures including WordNet-based, corpus based (LSA), literal-based (edit distance), Word2Vec based and monolingual alignment based have been employed as features for the supervised system. Except literal based measures, each feature individually has already led to a relatively good performance. However, a support vector regression model exploiting all features have performed best among others. The performance of the supervised system outperformed the winning system in SemEval 2015 Task 2 by a small margin. The idea of the study based on using a supervised approach with high-level features is similar to our approach, which will be described in Section 5.5.

We compared our biomedical domain specific systems with the domain-independent state-of-the-art systems ADW [68] and SEMILAR [69], both of which have online demos. In Chapter 6, the results of the ADW and SEMILAR tools are reported and compared with our systems. The similarity scores for each pair in our data set of ADW and SEMILAR tools have been obtained by using their online demos.

ADW [70] presented a unified semantic similarity approach performed on each level of text. It represents any lexical item (word, sentence or document) as a distribution over a set of word senses. Since WordNet ontology provides a rich network structure with different types of encoded relationships between concepts, it has been utilized to produce a frequency distribution for a lexical item. The idea is, random walks from multi-seeds produce a multinomial distribution over all the senses in WordNet. To construct each semantic signature, topic-sensitive PageRank [71] algorithm has been applied. This representation is called as item's semantic signature. To produce sense-based semantic signatures of lexical items, alignment based sense disambiguation algorithm is applied. This algorithm approaches the disambiguation task as an alignment problem. The semantic similarity calculation method in each level utilizes the produced semantic signatures. Various similarity measures including cosine, weighted overlap and jaccard have been used to compare semantic signatures. ADW has been evaluated on the data set provided by SemEval 2012 and the results have demonstrated

that it outperforms the top three ranking systems in SemEval 2012. Using a unified representation of text has yielded a state-of-the-art performance on different-levels of text including a sentence.

SEMantic simILARity software toolkit (SEMILAR) [72] is a toolkit that implements several semantic similarity measures based on WordNet and corpus statistics. Similar to other approaches, the sentence-level measures available through an online demo are based on mostly word-level similarities. SEMILAR offers different word-level similarity measures (LSA, WordNet-Lin) and algorithms (Optimal Matching [73], Greedy Pairing) to aggregate word-level similarities to compute sentence-level similarity score (Table 3.1). The LSA models have been developed on the TASA [74] corpus. As an ontology based word-level similarity measure, Lin algorithm is implemented and evaluated with different aggregation algorithms including Optimal Matching and Greedy Pairing. Optimal Lexical Matching is based on the optimal assignment problem which aims to find a maximum weight matching in a weighted bipartite graph, while Greedy Pairing matches each word in sentence S_1 with the word in sentence S_2 having the maximum similarity score. Moreover, they have provided the MCS algorithm [75] which proposes greedy approach based on word-to-word similarity measures.

Table 3.1. Available Measures through the SEMILAR Online Demo

Word-level Similarity Method	**Aggregation Algorithm**
WordNet-Lin	Optimal Matching
WordNet-Lin	Greedy Pairing
LSA	Optimal Matching
LSA	Greedy Pairing
MCS	Greedy Approach

4. DATA PREPARATION

Since there are no suitable data sets that comprise sentence pairs from the biomedical domain, we had to create our own test set. One of the major contributions of this study is to provide a gold standard data set of selected sentence pairs from biomedical literature which are manually annotated by human experts. Sentences are selected from the TAC [76] biomedical summarization track training data set, which is organized to develop technologies that aid in summarization of biomedical literature. TAC training data set consists of 20 papers and citing papers that vary from 12 to 20 for each of the reference papers. The corresponding reference text spans for each citing sentence are annotated by 4 different human experts and citing sentence-reference text span pairs can be found in the annotation files. Our selection process of citing sentences from these annotation files was performed by taking into consideration their text spans.

Our motivation to use TAC data set was that both semantically related and irrelevant sentence pairs occur in the annotation files. Some of the citing sentences cite the same reference papers because of similar reasons such as referring to a recent study on protein-protein interactions. These two citing papers mostly capture the same meaning linking them in a semantic way. On the other hand, there are also some citing sentences that cite the reference papers that are written about different research fields (one refers to study on microbiology, other mentions research on embrylogy). Therefore it was possible to obtain pair instances with a different range of similarity degrees by using this data set. The intuition behind the selection process was that if the corresponding reference text spans of a sentence pair are exactly or mostly the same, it is more than likely that the sentence pair is semantically very similar to each other. Moreover, if the reference papers of a sentence pair are different, probably, they are dissimilar to each other. Following this way of evaluation, possibly highly similar and very dissimilar sentence pairs are selected. Then, other instances possibly having moderate similarity degrees are chosen randomly from the citing sentences so that moderate similarity is the most common degree. After the selection process of 100 sentence pairs, each pair was annotated with a semantic similarity score independently

by five human experts. The similarity scores range between 0 and 4, where 4 refers to the highest semantic similarity and 0 is given for sentence pairs on irrelevant topics. The instruction set that is provided for annotators to guide how they will score the similarity between sentences for each similarity degree are shown in Table 4.1.

Table 4.1. Annotation Guideline

Annotation Score	**Definition**
0	The two sentences are on different topics.
1	The two sentences are not equivalent, but are on the same topic.
2	The two sentences are not equivalent, but share some details.
3	The two sentences are roughly equivalent, but some important information differs/missing.
4	The two sentences are completely or mostly equivalent, as they mean the same thing.

For defining these instructions, we were inspired from the guideline of SemEval 2012 Task 6 [58] on semantic textual similarity. Besides these instructions, example sentences are provided from biomedical literature for each of the similarity degree. These example sentence pairs that are scored between 0-4 are shown in Table 4.2. The collected data set format is shown in Figure 4.1.

Pair_ID | Sentence 1 | Sentence 2| Similarity_Score | Annotator_ID

Figure 4.1. Collected Data Set Format

Pair ID is unique for each pair in our data set. Sentence1 and Sentence2 are the sentence pair for the given Pair_ID. Similarity_Score is the score for the corresponding

Pair_ID given by the corresponding Annotator_ID. In our data set, Annotator_IDs for 5 different human annotators are symbolized as A, B, C, D, and E. One of the annotators, who is shown as D is a biomedical expert, while the other 4 annotators are from computer science.

The distribution of the scores by each of the annotators is shown in Figure 4.2. By considering this, it can be said that there are enough instances for each of the similarity degrees in our data set.

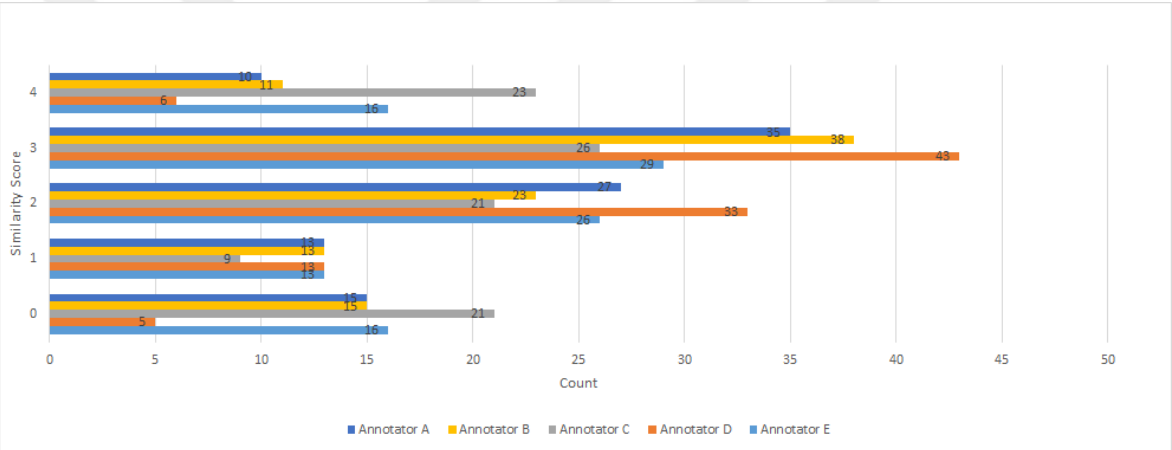


Figure 4.2. Distribution of the Similarity Scores in the Data Set

Table 4.2. Example Annotations

Sentence 1	**Sentence 2**	**Score**
Here we show that both C/EBPα and NFI-A bind the region responsible for miR-223 upregulation upon RA treatment.	Isoleucine could not interact with ligand fragment 44 (LF44), which contains amino group.	0
Membrane proteins are proteins that interact with biological membranes.	Previous studies have demonstrated that membrane proteins are implicated in many diseases because they are positioned at the apex of signaling pathways that regulate cellular processes.	1
This article discusses the current data on using anti-HER2 therapies to treat CNS metastasis as well as the newer anti-HER2 agents.	Breast cancers with HER2 amplification have a higher risk of CNS metastasis and poorer prognosis.	2
We were able to confirm that the cancer tissues had reduced expression of miR-126 and miR-424, and increased expression of miR-15b, miR-16, miR-146a, miR-155, and miR-223	A recent study showed that the expression of miR-126 and miR-424 had reduced by the cancer tissues.	3
Hydrolysis of B-lactam antibiotics by B-lactamases is the most common mechanism of resistance for this class of antibacterial agents in clinically important Gram-negative bacteria.	In Gram-negative organisms, the most common B-lactam resistance mechanism involves b-lactamase-mediated hydrolysis resulting in subsequent inactivation of the antibiotic.	4

5. METHODOLOGY

In this chapter, we will introduce our novel approaches on semantic sentence-level similarity in the biomedical domain. Our measures can be categorized into three types: distributional semantic vector models, ontology-based measures, and supervised approaches. Firstly, we will briefly explain our preprocessing method, which is applied for cleaning the text pair given to our semantic similarity systems. Then, we will provide the string similarity methods that have been evaluated on the data set. Finally and most importantly, we will introduce our solutions for the semantic sentence-level similarity problem in the biomedical domain in detail.

5.1. Preprocessing

Preprocessing is the first stage of all measures introduced in this chapter. Simple preprocessing steps consisting of removal of the punctuation marks and stop-words are applied to sentence pairs. Dictionary for stop-words was obtained from the default English stop-words list available online [77]. Moreover, it was modified by adding the frequently used abbreviation 'et al.' in sentences selected from academic papers. With the help of this dictionary, any word determined as stop-word is removed from the sentence.

The punctuation marks removed from the sentence pairs are listed below:

- Dot
- Comma
- Colon
- Exclamation Mark
- Semicolon
- Opening and Closing Parenthesis
- Square Brackets
- Question Mark

- Asterisk
- Underscore Character
- Slash Mark
- Dash

Our last preprocessing step was to lower-case sentence pairs. Then, sentences preprocessed in this module are given to the semantic similarity calculation modules as inputs.

5.2. String Similarity Methods

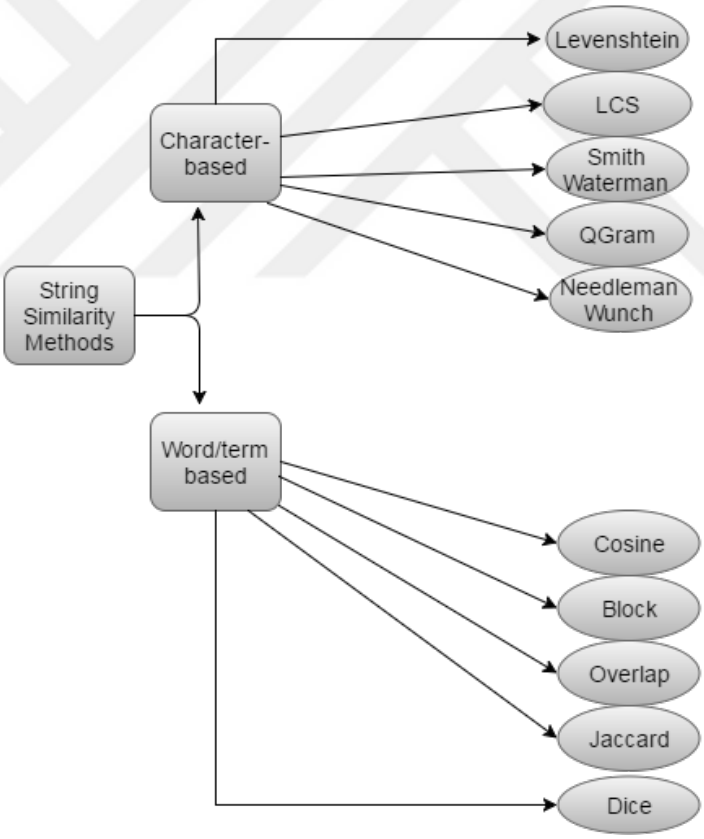


Figure 5.1. String Similarity Measures

We have evaluated well-known string similarity measures on our data set, since they are strikingly strong for estimating the semantic similarity between two strings. As shown in Figure 5.1, five character based and five term based similarity algorithms implemented in the SimMetrics package [78] have been evaluated.

5.3. Distributional Vector Models

Distributional semantic models learn the required information such as semantics, meaning for a given text from a large corpus. These measures enable semantic similarity systems to be easily adaptable to other domains, since they exploit knowledge from corpus statistics. In this study, we have tried out two distributional models introduced in the following sections in detail.

5.3.1. Latent Semantic Analysis

Latent Semantic Analysis (LSA) [79] is a method assuming that similar words occur in similar pieces of texts. It uses large collection of documents to construct semantic vectors. Firstly, the algorithm constructs a word-by-document matrix, where each row denotes a single unique word in the documents and the columns represent the documents in the collection. The corresponding value in the intersection of row r and column c, is assigned according to the umber of times a word is seen in the given document. Then Singular Value Decomposition (SVD) is calculated for the produced word-by-document matrix. As a result of this algorithm, three matrices $U\Sigma V$, where U corresponds to wordspace, V to document space, and Σ to the singular values, are produced. Then, the columns of U are truncated to the number of dimensions, which produces the final semantic vectors. The main aim of the SVD algorithm is to reduce the dimensional representation of the matrix in order to remove the noise and emphasize the strong relationships.

The LSA algorithm is mostly used on large documents. In this study, we have employed LSA to build 100 dimensional semantic vectors for each sentence. Then, the cosine similarity of these semantic vectors are considered as the similarity score between two sentences. The S:Space [80] library which implements the LSA algorithm has been used for this task. Sentence vectors have been trained on randomly selected sentences from the Open Access Subset of PubMed Central (OAS) [81] with the following command shown in Figure 5.2.

```
java -jar lsa.jar -d corpus.txt my-lsa-output.sspace
```

Figure 5.2. Command for Training the LSA Vectors

For training the LSA vectors, a subset of the OAS data set (around 5 MB) has been used due to memory constraints. In this Equation, corpus.txt is the name of the training file, while my-lsa-output.sspace denotes the output file, which would contain the produced vectors of each sentence in the training file.

5.3.2. Paragraph Vector

Paragraph Vector [82] is an unsupervised learning algorithm that learns fixed-length vector representations for variable lengths of texts such as sentences and has been proposed as a strong alternative to the simple bag-of-words model. In contrast to the bag-of-words model, paragraph vector captures the semantics of texts and considers the word order. Experiments on text classification and sentiment analysis tasks in the paper [82] showed that paragraph vector is a competitive approach with state-of-the-art systems and it overcomes many of the weaknesses of the bag-of-words model. Due to being able to provide semantically strong representation of texts, paragraph vectors have been used in many studies such as word embeddings [83] and semantic textual similarity [67].

Paragraph vector approach is based on the previous study for learning word vectors [84]. Although the focus of their work is on representing texts and predicting the surrounding words in contexts, it can be applied to various fields.

In this study, we have utilized paragraph vectors for measuring the similarity between two sentences. We have trained paragraph vectors on a randomly selected

subset (around 2 GB) of OAS of PubMed Central consisting of more than a million documents. They have been trained to learn 100, 150, 200 dimensional vectors of sentences, respectively. Then, the cosine similarity using Equation 2.1 between the corresponding paragraph vectors of texts is considered as a semantic similarity score between these texts. The command shown in Figure 5.3 has been used to train the paragraph vectors.

```
time\ ./word2vec -train ../input.txt -output vectors.txt -cbow 0
-size 200 -window 100 -negative 5 -hs 0 -sample 1e-3  -threads 100
-binary 0 -iter 1 -min-count 1 -sentence-vectors 1
```

Figure 5.3. Command for Training the Paragraph Vectors

The definition of each parameter is explained below.

- output: name of the output file where the sentence vectors are saved.
- iter: number of iterations over the corpus.
- cbow: if the parameter is set to 0, continuous bag-of-words model (CBOW) is used, if it is set to 1, skip-gram model is used.
- min_count: the algorithm ignores all words with total frequency lower than this value.
- size: is the dimensionality of the feature vectors.
- negative: if 0, negative sampling is used, the value of the parameter for negative specifies how many "noise words" should be drawn. If set to 0, no negative sampling is used.
- hs: if 1, hierarchical softmax is used for model training. If set to 0, and negative is non-zero, negative sampling is used.
- window: is the maximum distance between the current and the predicted word within a sentence.
- sample: threshold for configuring which higher-frequency words are randomly

downsampled.

In our evaluations, we set the size parameter to 100, 150, and 200. The other parameters have not been changed for our runs.

5.4. Ontology-based Similarity Methods

In this section, we introduce two sentence-level semantic similarity systems based on WordNet ontology (namely WordNet-based Similarity Module) and UMLS ontology (namely UMLS-based Similarity Module). The idea behind these approaches is to utilize knowledge based word-level similarity measures to obtain similarity scores among sentences due to the fact that a sentence consists of a set of words. Therefore, word-level similarity measures presented in Section 2.2 are used as core components for both systems. Then, these word-level similarity scores are used for obtaining sentence-level scores.

In Sections 5.4.1 and 5.4.2, firstly, we express the overall system flow for WordNet-based Similarity Module and UMLS-based Similarity Module. Then, we give some details about the tools that we have used for measuring semantic similarity between words and express the exact parameters, options, and measures that are used. In Section 5.4.3, we express the algorithm commonly used by the two systems, which is about using the word-level similarity scores for the calculation of sentence-level similarity.

5.4.1. WordNet-based Similarity Module

WordNet-based Similarity Module basically takes two sentences to be compared as inputs and returns the semantic similarity score by exploiting WordNet for these given sentences. The design of the system is shown in Figure 5.4. In this module, each word in a sentence is assumed as concept in WordNet. Words are determined by tokenizing the sentences according to the space character. Then, a dictionary is constructed from the extracted set of words in the two sentences to be compared. By using this dictionary and utilizing the WordNet-based word-level similarity component,

the sentence-level similarity score for a given sentence pair is obtained. Sentence-level similarity module gives each possible pair to the word-level similarity component and considers only the maximum score returned by the word-level similarity component. As a result of this process between the two modules interacting with each other, vectors are produced for sentence pairs. The overall sentence-level similarity score is obtained with the cosine similarity score of the two sentence vectors. This system is evaluated by using different ontology based measures for WordNet based word-level similarity module. The method for the calculation of word-level similarities is given in Section 5.4.1.1. The detailed algorithm and a walk through example of constructing sentence vectors are given in Section 5.4.3.

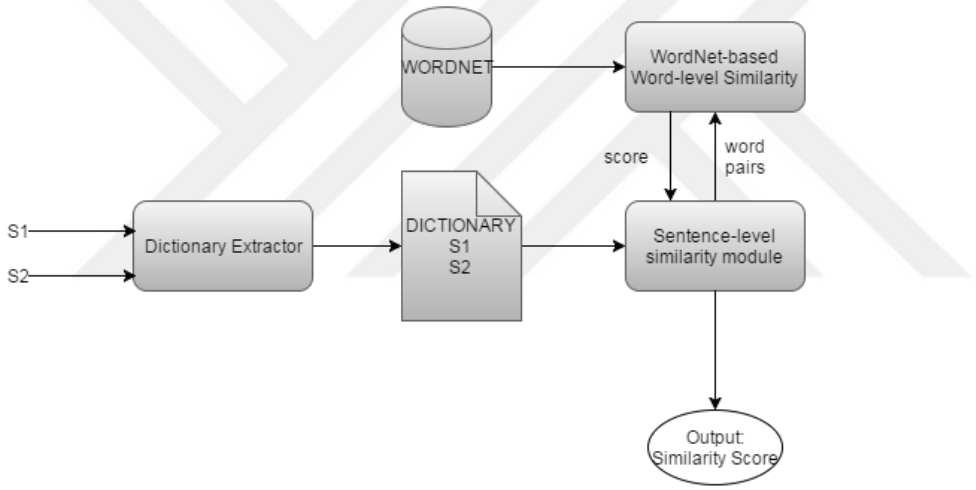


Figure 5.4. WordNet-based Similarity Method

5.4.1.1. WordNet-based Word-level Similarity. We have used WS4J library for calculation of the similarities between words utilizing the WordNet ontology. Word-level similarities are calculated by using the following algorithms implemented in WS4J.

- Resnik [23]
- Lin [24]
- Path [19]
- Jiang and Conrath [25]
- Leacock Chodorow [21]

- Wu and Palmer [22]
- Lesk [26]

These measures have been calculated using the Is-A relations in the WordNet ontology. Therefore, WordNet-based Word-level Similarity module is based on Is-A relations. As it is referred in the previous Section 2.3.1, the sense of a word is represented with the set of synonyms which is called synset and a word can be defined in multiple different syntactic categories. For example; the word 'love' is represented in the noun category, at the same time, it occurs in the verb category. So, matching the word with the correct node in the graph is challenging. We approached this problem as calculating the similarity between all possible pairs including synonyms and assigning the maximum similarity score. The pseudocode of the algorithm is given in Figure 5.5

```
maximumScore ⇐ 0
for all possible posTagPairs in posTagsList do
  synsetlist1 ⇐ getAllConceptsFor(word1, posTagPairs[1])
  synsetlist2 ⇐ getAllConceptsFor(word2, posTagPairs[2])
  for all synset1 in synsetlist1 do
    for all synset2 in synsetlist2 do
      score ⇐ similarityBetween(synset1, synset2)
      if score > maximumScore then
        maximumScore ⇐ score
      end if
    end for
  end for
end for
return maximumScore
```

Figure 5.5. Pseudocode for Computing Similarity between Word Pairs in WordNet

5.4.2. UMLS-based Similarity Module

Differently from the WordNet-based Similarity Module, UMLS-based Similarity Module uses METAMAP [85], which is a tool for extracting medical concepts from text rather than assuming each word as a concept. This approach is more reliable, since concepts can consist of more than one word. As shown in Figure 5.6, this process is named as Biomedical Concept Extractor. The METAMAP tool is run on both sentences S1 and S2 and a dictionary is constructed from the unique mapped concepts/phrases in the two sentences. Therefore, word-level similarity module utilizing UMLS takes concepts mapped by METAMAP as inputs. The rest of the methodology for constructing sentence-level vector is the same as the WordNet-based Similarity Module.

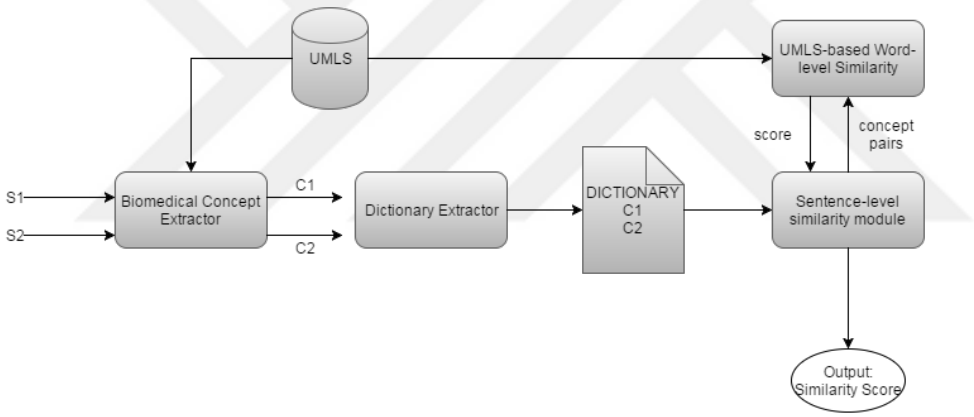


Figure 5.6. UMLS-based Similarity Method

5.4.2.1. UMLS-based Word-level Similarity. For the calculation of word-level similarities based on UMLS ontology, we have used the perl web interface of the UMLS:Similarity software package [36]. The main purpose of the interface is providing similarity information for CUIs. In this section, we express the sources and relationships available via the web interface in detail and which parameters are used for running the interface for the overall system. The similarity scores between concepts, which are calculated using the UMLS ontology, are used for measuring the semantic similarity between the whole sentences.

There are various semantic similarity and relatedness measures available via the web interface. To be able to obtain the similarity score, terms or CUIs associated to concepts in UMLS should be given as inputs to the interface. Moreover, the sources to be utilized and relation types to be used for measuring the similarity should be given as parameters. These available options are different for similarity and relatedness measures. Since relatedness is a more general concept, relation types and measures differ. The available sources and relations via the web interface for both similarity and relatedness are listed in Table 5.1.

Table 5.1. The available sources and relations via the UMLS:Similarity web interface

Measure	Source	Relations
SIMILARITY	OMIM, MeSH	PAR/CHD
	OMIM, MeSH	RB/RN
	OMIM	PAR/CHD
	OMIM	RB/RN
	MeSH	PAR/CHD
	MeSH	RB/RN
	SNOMEDCT	PAR/CHD
	SNOMEDCT	RB/RN
	FMA	PAR/CHD
	FMA	RB/RN
RELATEDNESS	SNOMEDCT	CUI/PAR/CHD/RB/RN
	SNOMEDCT	CUI
	UMLS_ALL	CUI/PAR/CHD/RB/RN
	UMLS_ALL	CUI
	MeSH	CUI/PAR/CHD/RB/RN
	MeSH	CUI

For similarity measures, relations which are PAR/CHD and RB/RN are hierarchical relation types. PAR/CHD represents parent-child relations, while RB/RN represents broader-narrower relations. For example, if the relation is chosen as PAR/CHD,

the similarity score is calculated using and considering the definition of the concept parent-child relations. On the other hand, for the relatedness measures, CUI refers to using the definition of the concept itself.

Through the UMLS:Similarity web interface, the semantic similarity scores between word pairs are calculated by using different similarity metrics. Moreover, our empirical analysis showed that most of the terms in our data set belong to either OMIM or MeSH ontology. As a result of this situation, the UMLS:Similarity web interface is restricted to search on the OMIM and MeSH ontologies. The different parameter settings for the calculation of word-pair similarity in our experiments are given in Table 5.2.

Table 5.2. Word-level UMLS Similarity

Measures	Sources	Relations
Resnik	OMIM, MeSH	PAR/CHD
Lin	OMIM, MeSH	PAR/CHD
JCN	OMIM, MeSH	PAR/CHD
Cdist	OMIM, MeSH	PAR/CHD
Path	OMIM, MeSH	PAR/CHD
LCH	OMIM, MeSH	PAR/CHD
Vector	UMLS ALL	CUI

Except the vector measure, all similarity metrics are evaluated by using the OMIM and MeSH ontologies and PAR/CHD relations. Since the vector metric measures the relatedness instead of similarity among concepts, the allowed sources, metrics and relations provided by the web interface are different. All similarity metrics employed by the UMLS:Similarity library are based on Is-A (PAR/CHD) relations. So, the proposed UMLS-based similarity module is based on Is-A relations. An example command for running the perl web interface is shown in Figure 5.7.

When the command is run, the web interface returns a similarity score between [0,1]. In the command, word1 and word2 are the word pairs to be compared. The option

```
query-umls-similarity-webinterface.pl --measure cdist --sab
OMIM,MSH --rel PAR/CHD word1 word2
```

Figure 5.7. Command for Running UMLS:Similarity Web Interface

measure defined the type of the measure, while the sab option shows the sources to be used for measuring similarity. Moreover, in this command, rel denotes the relation type to be used.

5.4.3. Algorithm for the Adaptation of Word Similarities to Sentence-level

A sentence consists of words. So, it is reasonable to use word-level similarities to compute sentence-level similarity. As mentioned in Sections 5.4 and 5.6, our ontology based measures exploit word-level similarities as their core component, then an algorithm is applied to construct sentence vectors by using these word-level similarities. In this section, we would like to give a walk-through example to make the applied algorithm clearer. For the algorithm illustrated below, we were inspired by the study [57].

5.4.3.1. A Walk-Through Example. The following is an illustration which introduces the algorithm and the required steps to build the semantic vector of a sentence.

- S1: Necroptosis requires the activity of RIP1 and RIP3.
- S2: Necroptosis could be induced via modulating RIP1 and RIP3.

The dictionary, D (union of the unique words from each sentence, S1 and S2) is: {Necroptosis, requires, the, activity, of, RIP1, and, RIP3, could, be, induced, via, modulating}. D is used to build semantic vectors for S1 and S2, which have the same dimensionality as the dictionary. For instance, in order to build a semantic vector for S1, each word in the dictionary is compared with every word in S1 and the highest

similarity score is assigned for the corresponding dimension index in the vector. As shown in Figure 5.9, D is obtained by using all distinct words in S1 and S2. For determining the score of the 10th dimension of semantic vector S1, similarity scores between each word in S1 and the 10th dimension of D are computed. Since the highest score is 0.33 among all similarity scores, the score of the 10th index of S1 is considered as 0.33. This process is repeated for the remaining part of semantic vector S1. Then, the same algorithm is applied to learn the semantic vector of S2. Finally, the cosine similarity between S1 and S2 gives the semantic similarity score between them. The pseudocode of the described algorithm is given in Figure 5.8.

Function constructSemanticVector(S, dictionary, ontology):
  $similarity_{max} \Leftarrow 0$
  $index \Leftarrow 0$
  for *all* $word_{Dictionary}$ *in dictionary* **do**
    for *all* $word_{Sentence}$ *in* S **do**
      $similarity \Leftarrow calculateSimilarity(word_{Dictionary}, word_{Sentence}, ontology)$
      if $similarity_{max} < similarity$ **then**
        $similarity_{max} \Leftarrow similarity$
      end if
    end for
    $SemanticVector_S[index] \Leftarrow similarity_{max}$
    $index \Leftarrow index + 1$
  end for
  return $SemanticVector_S$
EndFunction

Figure 5.8. Algorithm for Building a Semantic Vector of a Sentence

5.4.4. Combined Ontology Methods

In this study, our first motivation was that combined approaches are needed for semantic similarity computation in the biomedical domain, since sentences in biomedical literature consist of both general terms and biomedical-specific terms. In Sections

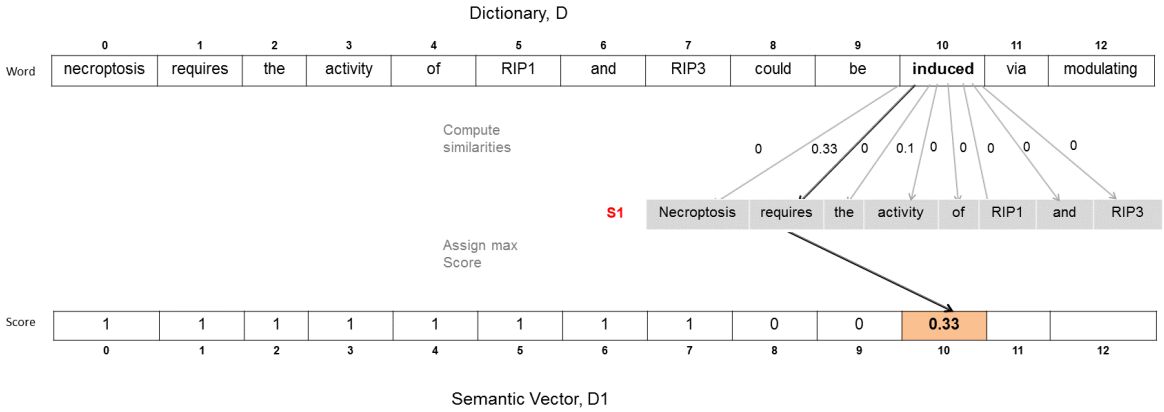


Figure 5.9. Illustration of the proposed algorithm which constructs semantic vectors of sentences

5.4.1 and 5.4.2, different sentence-level similarity methods exploiting either WordNet or UMLS ontology are introduced. To utilize the knowledge from both UMLS and WordNet ontologies, we propose a new approach and two different combination scenarios in this section. One of the combined methods is performed on sentence-level, while the other proposed approach combines ontology based systems on word-level. As a combination Equation, we have used:

$$combinedscore = Score_{WordNet}.\lambda + Score_{UMLS}.(1 - \lambda) \tag{5.1}$$

5.4.4.1. Word-level combined ontology method. Word-level combined ontology method performs the combination of two different methods on word-level. For computing similarity of word pairs in a sentence, the similarity score of the measure based on WordNet ontology and the UMLS ontology are combined using Equation 5.1. λ in the Equation is set to 0.5. Then, the algorithm described in Section 5.4.3 is applied to compute sentence-level similarity by using these combined word-level similarities.

5.4.4.2. Sentence-level combined ontology method. As shown in Figure 5.10, sentence level combined ontology method takes the similarity score of WordNet based similarity

module and UMLS based similarity module for a sentence pair, then combines these scores by using Equation 5.1. λ in the Equation is set to 0.1, 0.3, 0.5, 0.7, and 0.9, respectively. The pseudocode of sentence-level combined algorithm is given in Figure 5.11.

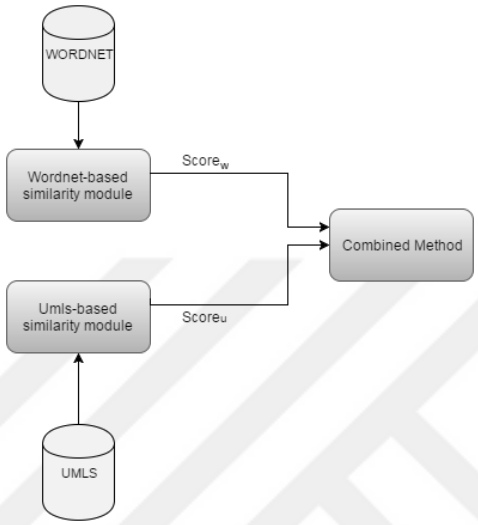


Figure 5.10. Sentence-level Combined Ontology Method

5.5. Supervised Approaches

In this section, we introduce our supervised systems. For comparing different regression algorithms for this task, learning models including SVR, Random Forest, Linear Regression, and MLP have been used. Each learning model has been evaluated with different feature sets. We categorized our features as low-level and high-level features. Detailed definition of feature sets are presented in the following Section 5.5.1.

5.5.1. Features

5.5.1.1. Low-level Features. N-gram Overlap: An n-gram is defined as contiguous n items from a text. An item can be a letter, a word or a phoneme. In our experiments, an item denotes a word. N-gram language models have been widely used in many applications in NLP. Since our problem is to determine the similarity score between

input: λ is combination parameter, $sentence_1$ and $sentence_2$ are sentence pairs

Function calculateCombinedSimilarity($sentence_1, sentence_2, \lambda$):

$S_1 \Leftarrow preprocess(sentence_1)$

$S_2 \Leftarrow preprocess(sentence_2)$

$dictionary_w \Leftarrow extractUniqueWords(S_1, S_2)$

$S_{bio1} \Leftarrow biomedicalConceptExtractor(S_1)$

$S_{bio2} \Leftarrow biomedicalConceptExtractor(S_2)$

$dictionary_u \Leftarrow extractUniqueWords(S_{bio1}, S_{bio2})$

$Vector_{S_1,WordNet} \Leftarrow constructSemanticVector(S_1, dictionary_w, wordnet)$

$Vector_{S_2,WordNet} \Leftarrow constructSemanticVector(S_2, dictionary_w, wordnet)$

$Vector_{S_1,UMLS} \Leftarrow constructSemanticVector(S_1, dictionary_u, UMLS)$

$Vector_{S_2,UMLS} \Leftarrow constructSemanticVector(S_2, dictionary_u, UMLS)$

$score_{wordnet} \Leftarrow cosineSimilarity(Vector_{S_1,WordNet}, Vector_{S_2,WordNet})$

$score_{UMLS} \Leftarrow cosineSimilarity(Vector_{S_1,UMLS}, Vector_{S_2,UMLS})$

$score \Leftarrow score_{wordnet} * \lambda + score_{UMLS} * (1 - \lambda)$

return score

EndFunction

Figure 5.11. Sentence-level Combined Ontology Method Pseudocode

two sentences, we have used n-gram overlap between two sentences as a feature.

Let S_1 and S_2 be the sets of consecutive n-grams in the first sentence and second sentence, respectively. The n-gram overlap is defined as:

$$Overlap_{ngram} = S_1 \cap S_2 \tag{5.2}$$

The overlap, defined in Equation 5.2, is computed for unigrams and bigrams. Bigram is a sequence of two adjacent elements in a string, while unigram is a single element in a string.

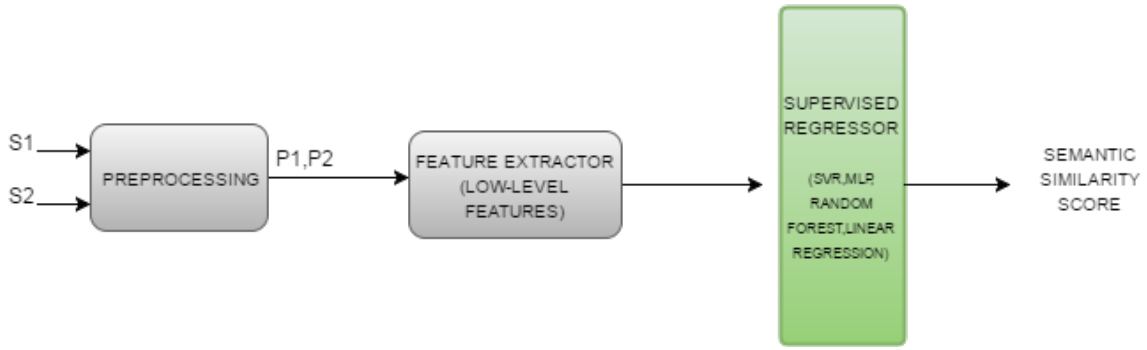


Figure 5.12. Supervised Similarity Method exploiting low-level features

We give an example on computing unigram and bigram overlaps across two sentences S_1 and S_2 listed below.

- S_1 : KRAS mutations are common in cancer.
- S_2 : KRAS are dependent on Craf.

The individual n-grams (unigram, bigram) of the sentences and the corresponding overlaps are computed as follows. 's' denotes the beginning and end of a sentence.

$$\begin{aligned}
\text{Bigrams}_{S_1} &= \{\text{(s, KRAS), (KRAS,mutations), (mutations,are),} \\
&\quad \text{(are,common),(common,in), (in,cancer), (cancer, s)}\} \\
\text{Bigrams}_{S_2} &= \{\text{(s, KRAS), (KRAS, are), (are,dependent),} \\
&\quad \text{(dependent,on),(on,Craf), (Craf,s)}\} \\
\text{Common bigrams}_{S_1,S_2} &= \{\text{(s, KRAS)}\} \\
\text{Unigrams}_{S_1} &= \{\text{(KRAS), (mutations), (are), (common), (in) ,(cancer)}\} \\
\text{Unigrams}_{S_2} &= \{\text{(KRAS), (are), (dependent), (on), (Craf)}\} \\
\text{Common unigrams}_{S_1,S_2} &= \{\text{(KRAS), (are)}\}
\end{aligned}$$

5.5.1.2. High-level Features. As high-level features, we considered using the similarity scores of the unsupervised systems selected from each category described in Sections 5.2, 5.4 and 5.3. By using this approach, Qgram as string similarity approach,

paragraph vector as distributional vector model, and sentence-level combined ontology method as ontology-based system have been used as high-level features. The results obtained by these systems are presented in Chapter 6. The supervised system exploiting high-level feature sets is illustrated in Figure 5.14. Preprocessed sentences are given to each unsupervised system as inputs. Then, the output score of each system, which is the semantic similarity score for the given pair, is used as feature in our supervised system. The pseudocode of the high-level supervised approach is given in Figure 5.13.

input: features(3) is the 3 dimensional vector which saves extracted feature in each dimension
Function featureExtractor(S_1, S_2):
    $features(1) \Leftarrow calculateCombinedSimilarity(S_1, S_2, 0.5)$
    $features(2) \Leftarrow calculateParagVecSimilarity(S_1, S2)$
    $features(3) \Leftarrow calculateQGramSimilarity(S_1, S_2)$
    return features
EndFunction
input: trainModel is model learned by regression algorithm using training data set.$sentence_1$ and $sentence_2$ are a sentence pair.
$S_1 \Leftarrow preprocess(sentence_1)$
$S_2 \Leftarrow preprocess(sentence_2)$
$testFeatures \Leftarrow featureExtractor(S_1, S_2)$
$score \Leftarrow evaluate(trainModel, testFeatures)$
return score

Figure 5.13. Pseudocode of the High-level Supervised Approach

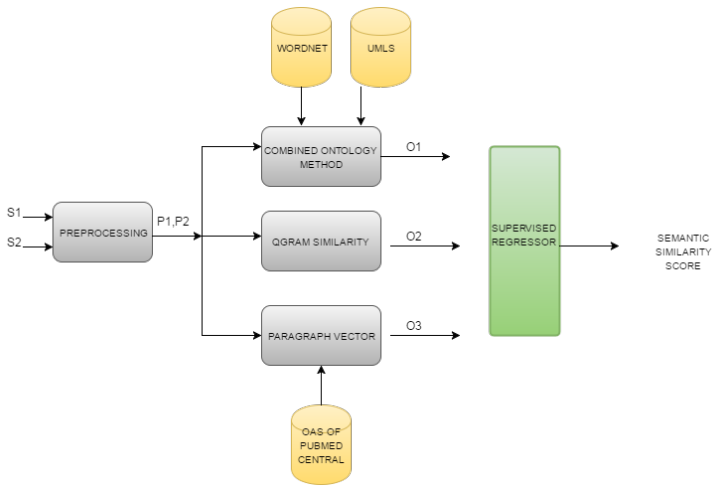


Figure 5.14. Supervised Similarity Method Exploiting High-level Features

6. EXPERIMENTS AND RESULTS

In this section, we report and discuss the results of the approaches presented in Chapter 5. Since there is no previous study on sentence semantic similarity computation in the biomedical domain, there were no suitable baselines to indicate the actual efficiency of our solutions. Therefore, we considered the domain-independent state-of-the-art approaches ADW and SEMILAR introduced in Section 3.2 as our baseline models and evaluated the performance of all measures by comparing with these baseline systems. The sentence pairs were annotated by five different human experts in this study which enabled the test data to be more robust and reliable. All reported correlation results of the measures in this section were computed by considering ground truth (GT) as the mean of annotator scores, since there was not a certain ground truth. Now, we will explain our performance metric in the following Section 6.1, then we will report the results of the proposed approaches.

6.1. Evaluation Metric

In order to evaluate our systems, we have used the Pearson correlation metric. Pearson correlation [86] is a metric defined by Equation 6.1 that measures the linear correlation between X and Y. Its value ranges between -1 and +1, with 1 meaning that two variables are totally correlated, -1 denoting a total negative correlation and 0 meaning that these two variables have no correlation.

$$r = \frac{\sum XY - \frac{\sum X \sum Y}{N}}{\sqrt{\sum X^2 - \frac{(\sum X)^2}{N}}\sqrt{\sum Y^2 - \frac{(\sum Y)^2}{N}}} \tag{6.1}$$

The strength can be assessed by the general guideline proposed in the study [87] by Evans to interpret effect size. Association strength is defined as listed below:

- `very strong` : 0.80-1.00
- `strong` : 0.60-0.79

- `moderate` : 0.40-0.59
- `weak` : 0.20-0.39
- `very weak` : 0.00-0.19

Correlation between two variables is defined as very strong, strong, moderate, weak or very weak according to the score calculated by using Equation 6.1. In this study, it is aimed to reach very strong relationship with the mean score of annotators.

We have determined our lower bound as the correlation coefficients of the ADW and SEMILAR toolkits with the ground truth. The other factor which should be taken into consideration is determining the upper bound that could be expected from an algorithmic measure. We have followed the same approach with the research [57] to determine the upper bound of the study. We calculated the correlation coefficient for the judgments of each participant against the rest of the group and then took the mean. The results are shown in Table 6.1. In this table, the biomedical expert annotator, who is symbolized as Annotator D, obtained the lowest correlation 0.902, which is still quite high.

Table 6.1. Correlation scores among annotators

	Correlation r
Annotator A	0.952
Annotator B	0.958
Annotator C	0.917
Annotator D	0.902
Annotator E	0.941

6.2. Results

In this section, the correlation results of all proposed approaches and baseline models on our test set are reported and discussed. To be able to observe the effect of preprocessing methods on semantic similarity measures, we have evaluated our mea-

sures using three different techniques:

(i) Sentence pairs without any preprocessing.
(ii) Sentences with the removal of punctuations.
(iii) Pairs with the removal of punctuations and stop-words.

ADW and SEMILAR have been evaluated through their online demos. There are different measures available through SEMILAR online demo. We have chosen WordNet Lin measure to calculate word-level similarities and optimal matching algorithm to aggregate word-pair similarities. For ADW, we set 'Alignment Based Disambiguation' algorithm as 'Yes'. In Table 6.2, the correlation of domain independent systems namely ADW and SEMILAR with GT (Ground Truth) are presented. Both ADW and SEMILAR have shown moderate correlation. This poor results actually demonstrated the need of new approaches for this domain-specific research field. Since these state-of-the-art systems exploit only WordNet ontology, poor results for this task is reasonable. To adapt this methods for biomedical domain, they both can use the UMLS ontology as well as WordNet. Since ADW performed significantly better than SEMILAR, we have considered ADW as our baseline model.

Table 6.2. Correlation scores for domain-independent state-of-the-art systems

	Correlation r
ADW	0.586
SEMILAR	0.419

6.2.1. String Similarity Measures

As discussed in Section 5.2, we have evaluated five character-based and five term-based string similarity measures on our data set. The correlation scores of each string similarity measure with GT are presented in Table 6.3. For each measure, three results were reported according to the level of preprocessing performed. The first result

is obtained without any preprocessing approach on the sentence pairs. Experiments showed that the application of preprocessing methods have contributed significantly to the performance of string similarity measures. The range of increase for accuracy varies between 10% and 31%. This result is reasonable and expected since string-based approaches are highly sensitive to small changes, since they do not take into consideration semantic information of text. Moreover, we observed that term-based similarity approaches are slightly better than character-based measures. All term based similarity measures (block, jaccard, cosine, overlap and dice) have obtained strong Pearson correlation with GT, while some character based measures such as Needleman Wunch had moderate correlation. However, the best correlation among the string based approaches has been obtained by Qgram similarity, which is a character based measure.

In summary and according to the results, it can be said that string similarity methods perform strikingly well and are very sensitive to any changes performed by preprocessing methods. Qgram similarity, the best performing string similarity method, has been used as one of the high-level features for the supervised system that will be mentioned in Section 5.5.

6.2.2. Distributional Semantic Vector Models

Table 6.4 shows the Pearson correlation coefficients between each distributional vector model and GT. Since paragraph vectors were trained to learn 100, 150 and 200 dimensional vectors of sentences, each model having different dimensions are reported. Paragraph vector has shown remarkable performance, which is not much affected by the size of the vector. These results showed the effectiveness of the paragraph vector method for semantic vector representation of short texts. However, LSA which is trained to learn 100 dimensions of vectors had very weak correlation. The weak performance of LSA for sentence-level semantic similarity has also been discussed in the study [57].

LSA exploits co-occurrence information of words. The idea behind this is that similar words occur in similar contexts. However, in short texts such as sentences,

Table 6.3. Results of the String Similarity Measures

	Preprocessing	**Correlation r**
Cosine	no-preprocessing	0.458
	removal punctuations	0.575
	+ removal stop-words	0.724
Jaccard	no-preprocessing	0.569
	removal punctuations	0.710
	+ removal stop-words	0.710
Block	no-preprocessing	0.558
	removal punctuations	0.696
	+ removal stop-words	0.752
Levenshtein	no-preprocessing	0.504
	removal punctuations	0.533
	+ removal stop-words	0.592
LCS	no-preprocessing	0.575
	removal punctuations	0.593
	+ removal stop-words	0.595
Needleman Wunch	no-preprocessing	0.411
	removal punctuations	0.429
	+ removal stop-words	0.444
Overlap Coefficient	no-preprocessing	0.564
	removal punctuations	0.695
	+ removal stop-words	0.695
Smith Waterman	no-preprocessing	0.640
	removal punctuations	0.660
	+ removal stop-words	0.675
Qgram	no-preprocessing	0.677
	removal punctuations	0.731
	+ removal stop-words	**0.754**
Dice	no-preprocessing	0.398
	removal punctuations	0.709
	+ removal stop-words	0.709

Table 6.4. Correlation Scores for Paragraph Vector and LSA

Method	Vector Size	Correlation r
Paragraph Vector	100	**0.787**
	150	0.783
	200	0.786
LSA	100	0.158

word co-occurrence may be rare. People may express similar meanings using different sentences in terms of word content. These situations cause LSA to be less effective for sentences.

6.2.3. Ontology-based Similarity Systems

Results illustrated in Table 6.5 has shown that our WordNet based measures have been affected from preprocessing methods. This result was expected since we assumed each word as a concept in WordNet for Wordnet-based measures. So, if a word does not refer the exact concept because of small lexical differences, word-level algorithm can not recognize it. WordNet-based similarity module using path algorithm as word-level similarity approach yield the best performance with the 0.644 Pearson correlation score among others.

The correlation results for UMLS-based similarity measures exploiting several different word-level algorithms have been presented in Table 6.6. An observation is that removal of punctuations does not affect the performance of the UMLS based system. This result is reasonable since UMLS based similarity module uses the METAMAP tool for extracting medical terms and METAMAP can handle punctuations. Using METAMAP enables the system to be more robust to small changes. Similarly to WordNet-based similarity module, the path algorithm used for word-level similarity performed better than the others. Although the path algorithm based on the length of the path between two concepts is the simplest word-level ontology-based approach

Table 6.5. Correlation scores for WordNet based similarity systems

	Preprocessing Methods	**Correlation r**
WordNet-Path	no-preprocessing	0.487
WordNet-Path	removal punctuations	0.600
WordNet-Path	+removal of stop-words	**0.644**
WordNet-Resnik	no-preprocessing	0.159
WordNet-Resnik	removal punctuations	0.192
WordNet-Resnik	+removal of stop-words	0.234
WordNet-WP	no-preprocessing	0.245
WordNet-WP	removal punctuations	0.374
WordNet-WP	+removal of stop-words	0.354
WordNet-Lin	no-preprocessing	0.401
WordNet-Lin	removal punctuations	0.504
WordNet-Lin	+removal of stop-words	0.495
WordNet-Lesk	no-preprocessing	0.339
WordNet-Lesk	removal punctuations	0.377
WordNet-Lesk	+removal of stop-words	0.400
WordNet-JCN	no-preprocessing	0.448
WordNet-JCN	removal punctuations	0.486
WordNet-JCN	+removal of stop-words	0.623
WordNet-LCH	no-preprocessing	0.121
WordNet-LCH	removal punctuations	0.266
WordNet-LCH	+removal of stop-words	0.287

Table 6.6. Correlation scores for UMLS based similarity systems

	Ontology	**Relation**	**Preprocessing Methods**	**Correlation**
Vector	UMLS	CUI	no-preprocessing	0.325
			removal punctuations	0.325
			+removal stop-words	0.414
Cdist	OMIM-MeSH	PAR/CHD	no-preprocessing	0.578
			removal punctuations	0.578
			+ removal stop-words	0.650
Lin	OMIM-MeSH	PAR/CHD	no-preprocessing	0.556
			removal punctuations	0.556
			+ removal stop-words	0.645
WP	OMIM-MeSH	PAR/CHD	no-preprocessing	0.504
			removal punctuations	0.504
			+ removal stop-words	0.576
JCN	OMIM-MeSH	PAR/CHD	no-preprocessing	0.580
			removal punctuations	0.580
			+ removal stop-words	0.624
LCH	OMIM-MeSH	PAR/CHD	no-preprocessing	0.222
			removal punctuations	0.222
			+ removal stop-words	0.333
Resnik	OMIM-MeSH	PAR/CHD	no-preprocessing	0.461
			removal punctuations	0.461
			+ removal stop-words	0.473
Path	OMIM-MeSH	PAR/CHD	no-preprocessing	0.605
			removal punctuations	0.605
			+ removal stop-words	**0.651**

among others, both sentence-level UMLS-based and WordNet-based similarity methods have obtained the best correlation scores by using the path measure on our sentence pair data set.

In general, although the individual accuracy of the ontology-based models are reasonable enough, the increase in the correlation performance for all combined models, compared to the individual correlation scores indicate that the combination is useful. The results of the combined approaches shown in Table 6.7 justify our hypothesis which was based on exploiting two ontologies for domain-specific literature. When we examine the results, it is seen that although word-level combined approach performed better than the individual approaches, sentence-level combination scenarios achieved the best performance. In particular, using λ parameter as 0.5 in other words averaging the individual WordNet-based path (0.644) and UMLS-based path (0.651) measures has increased the performance significantly with the correlation score 0.710.

Table 6.7. Correlation scores for combined ontology method

Method	λ	Correlation r
Sentence-level Combined Method	0.1	0.671
	0.3	0.699
	0.5	**0.710**
	0.7	0.699
	0.9	0.666
Word-level Combined Method	0.5	0.668

6.2.4. Supervised Approaches

We have presented two different supervised approaches which are based on low-level and high-level features in Section 5.5. In this section, we discuss the results of each supervised approach. Among low-level features, the best result is obtained by using the Random Forest as a learning model and unigram overlap as feature. However, even the

best result among learning models using n-gram features indicated a weak correlation with the ground truth. Experiments showed that using n-gram overlaps as features causes poor performance.

To combine the best performing measures of each category namely distributional vector models, string similarity measures, ontology-based approaches, supervised learning algorithms have been used. Best performing systems in each category are listed below.

(i) Qgram Distance (String Similarity Measure)
(ii) Paragraph Vector (Distributional Semantic Vector Model)
(iii) Sentence-level Combined Method [$\lambda = 0.5$](Ontology-based Method)

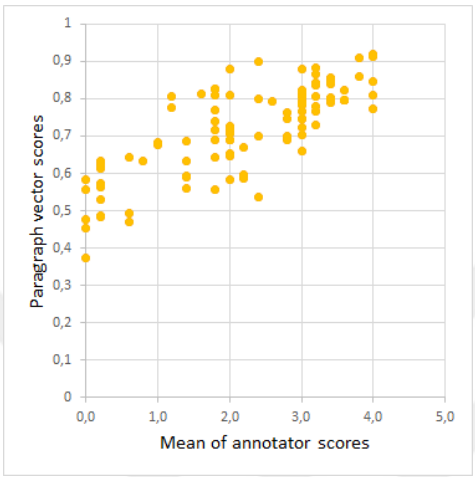
The similarity score of each unsupervised method for a sentence pair has been used as high-level feature. As learning models, MLP, Random Forest, and, Linear Regression implemented in Weka [88] and SVR implemented in LibSVM [89] referred to in Section 2.4 have been employed. For building a Linear Regression model, we have used Weka Java library and did not need any extra parameters. Number of trees for Random Forest was determined as 10. The default parameters listed below were used for MLP and SVR.

- MLP: -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H 4
  (L: learning rate for backpropagation algorithm, M: Momentum Rate for the backpropagation algorithm, N: Number of epochs to train through, V: Percentage size of validation set to use to terminate training, S: The value used to seed the random number generator, E: The consecutive number of errors allowed for validation testing before the network terminates, H: The hidden layers to be created for the network)
- SVR: -s 4 -t 0
  (s: svm type [4 denotes nu-svr model], t: kernel type [0 indicates linear kernel type])

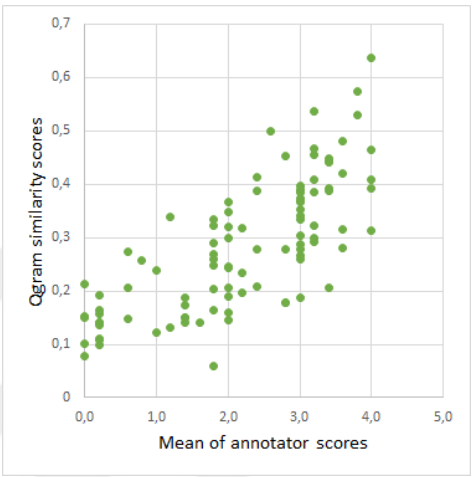
Evaluations for all the supervised models are performed using stratified 10-fold cross-validation over all the sentence pairs due to the small size of data set. Before building a learning model, instances were randomized then, 10-fold cross-validation has been applied. The final results for supervised semantic similarity systems are obtained by averaging the individual correlation results of each fold. The experimental results indicated that combining high-level features significantly increases the individual performance of each unsupervised system. This means that these unsupervised system scores complement each other in some way. The correlations between each unsupervised system used for supervised learning as high-level feature and GT are shown in Figures 6.1(a), 6.1(b) and 6.1(c). Although each unsupervised method had strong association with GT, combination of these approaches by supervised algorithm has led to very strong correlation. Since Linear Regression exploiting high level features has shown the best performance 83.6% among others, we considered it as our model for supervised system. The results presented in Figure 6.1(d) were reported according to the Linear Regression model.

Table 6.8. Correlation scores for supervised similarity system

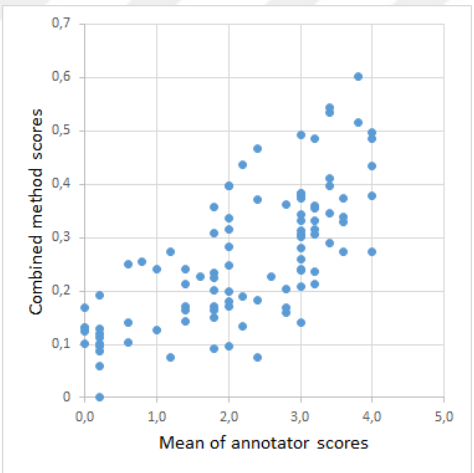
Features	**LR**	**RF**	**SVR**	**MLP**
Unigram Overlap	0.362	0.397	0.064	0.226
Bigram Overlap	0.366	0.357	0.094	0.081
ParagVec+combined method	0.807	0.782	0.679	0.817
ParagVec+Qgram	0.831	0.797	0.717	0.832
ParagVec+combined method+Qgram	**0.836**	0.817	0.709	0.829



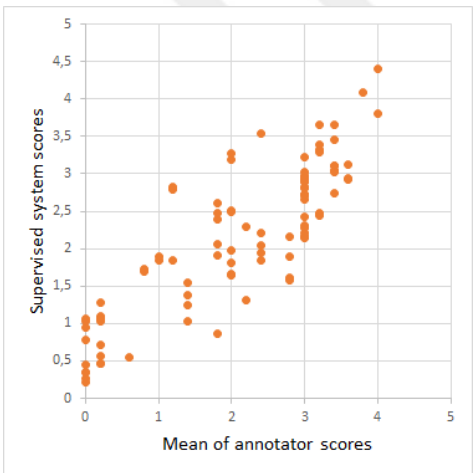
(a) Correlation between paragraph vector and ground truth



(b) Correlation between qgram measure and ground truth



(c) Correlation between combined system and ground truth



(d) Correlation between supervised system and ground truth

Figure 6.1. Correlation Between Systems and Ground Truth

7. CONCLUSION AND FUTURE WORK

7.1. Conclusions

In this study, we adapted and presented several approaches and methodologies for semantic sentence similarity. One of the most remarkable contributions to the academic literature of this study is to demonstrate the need of adapted or new approaches for semantic sentence-level similarity in the biomedical domain. With this work, the inadequacy of domain-independent semantic similarity measures for a domain-specific problem has been illustrated. Another important contribution is that this research provides a strong baseline as well as a hand-crafted data set for further studies due to attempting the first methods on this unstudied research area.

We can categorize our approaches into three main areas; distributional vector models, ontology-based methods, and supervised approaches. Paragraph vector and LSA which both represent the sentences as vectors capturing semantics have been evaluated on our data set. We have revealed that LSA is not very effective for sentences. However, paragraph vector, which has been proposed as an alternative to the bag-of-words model, achieved a quiet remarkable performance by showing strong correlation with the ground truth scores. This result has shown that paragraph vector is a good method on representing sentences as vectors capturing semantics.

Thanks to the ontologies that enable the computation of semantic distances between concepts, ontology-based measures have been used for our semantic similarity problem. Since sentences are selected from biomedical papers in our data set, we have utilized WordNet as a general domain ontology and UMLS as a biomedical domain specific ontology. Popular ontology-based algorithms, which calculate similarity between concepts, have been employed for WordNet and UMLS, respectively. The most important finding from the evaluations of ontology-based approaches was that combined approaches using both WordNet and UMLS ontology rather than single ontology are more prospering on estimating the similarity between biomedical sentences.

This result is reasonable since sentences consist of both biomedical and general concepts and probably a word in a sentence occurs either in WordNet or UMLS. In other words, knowledge coming from both WordNet and UMLS complements each other and contributes to the overall performance of the system.

Besides, popular string similarity measures including term-based and character-based methods have been evaluated on the our data set. Experiments showed that simple lexical similarity measures perform strikingly good on the semantic sentence similarity task.

Finally, we have presented a supervised semantic similarity estimation system which exploits high-level features. High-level features consist of the similarity scores of the best performing systems in each category. Combining unsupervised methods with the help of supervised learning models has increased the overall performance of the system. As learning models, different regression algorithms have been used. Linear regression with high-level features has obtained the best performance among all proposed systems. Experiments showed that each system using different approach to estimate the similarity contributes to the overall performance of the system.

One of the important contributions of this study is that providing a test set for researchers and future studies on semantic sentence similarity in the biomedical domain. Selected 100 sentence pairs were annotated by five human experts for this research and all measures were evaluated on this crafted data set.

To sum up, we have addressed the semantic sentence-level similarity problem in the biomedical domain and attempted the first adapted approaches in this research field. Among several measures, the supervised system exploiting high-level features achieved the strongest Pearson correlation of 83.6% with ground truth scores. We believe that our biomedical-domain specific semantic sentence-level similarity measure can be used in various applications of BioNLP such as automatic summarization, question answering, text categorization, and text retrieval in biomedical literature. However, another factor that should be taken into consideration is the upper bound that could

be expected from an algorithmic measure. Upper bound in this study can be considered as the performance of a typical human, which is 90.2% according to the lowest correlations between human annotators. So, we can say that there is still room for improvement on biomedical domain specific semantic sentence similarity measures.

7.2. Future Studies

Since we adapted previous domain independent approaches on semantic similarity computation to biomedical domain, there is open room for new algorithms and methodologies developed specifically for the biomedical domain. In future work, we aim to focus on new algorithms for semantic similarity computation in the biomedical domain.

Moreover, the following can be performed in order to improve our adapted ontology based algorithm. Several popular ontology based word-level similarity measures have been used to calculate distances between words for our sentence-level similarity measures. So, these algorithms and their performances have direct effect on the overall performance of the ontology-based sentence-level similarity systems. All measures used in this study have their own drawbacks. Various adaptations and hybrid approaches were proposed to overcome the weaknesses of these ontology-based word-level similarity metrics (referred in Chapter 3). In order to leverage the word similarity measures, which is the core problem in our ontology-based approaches, adapted hybrid approaches can be experimented instead of simple measures. A more effective word-level measure would yield a more effective sentence-level similarity system. On the strength of this hypothesis, we will try hybrid approaches for the calculation of word-level similarity in the near future.

Furthermore, we believe that the following suggestions are worth trying to evaluate our proposed measure as future studies.

(i) Recently, citation based summarization has drawn the attention of the researches, since citing sentences tend to contain important information about the reference

paper. Citation based summarization methods tackle the problems of detecting the sentences that cite the reference paper (i.e., citing sentences), selecting the most salient ones, and producing a structured summary. Although using citations is useful for the summarization task, there is still room for improvement, since citation sentences might not accurately represent the content of the cited article, as they often fail to capture the context of the reported findings. According to TAC intuition, the corresponding reference text span in the reference paper can give more detailed and better expression about the important aspects of the reference paper. And, TAC has organized a task to address this problem consisting of two sub-tasks for biomedical domain. A study [90] has been recommended as a solution of this problem. However, this study approaches the problem as a retrieval task. We believe that addressing the problem of matching citation text and cited spans by using semantic similarity methods is quite reasonable. Therefore, our semantic similarity system for biomedical domain can be useful as a solution this problem.

(ii) LexRank [91] is a graph-based method proposed for extractive summarization, which aims to find the most informative sentences in text. It computes sentence importance based on the concept of eigenvector centrality in a graph representation of sentences. This model exploits inter-sentence similarity as one of the major part of the algorithm. However, it uses simple cosine similarity metric to obtain the similarity degree between sentences and does not exploit any semantic features of texts. We believe that our sentence-level semantic similarity measure developed for biomedical domain models inter-sentence similarities in more effective way and may help better detection of salient sentences in text.

REFERENCES

1. Jeon, J., W. B. Croft and J. H. Lee, "Finding similar questions in large question and answer archives", *Proceedings of the 14th ACM international conference on Information and knowledge management*, pp. 84–90, ACM, 2005.

2. Chan, Y. S. and H. T. Ng, "MAXSIM: A Maximum Similarity Metric for Machine Translation Evaluation.", *ACL*, pp. 55–62, 2008.

3. Finch, A., Y.-S. Hwang and E. Sumita, "Using machine translation evaluation techniques to determine sentence-level semantic equivalence", *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, pp. 17–24, 2005.

4. Wang, D., T. Li, S. Zhu and C. Ding, "Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization", *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 307–314, ACM, 2008.

5. Wang, H., L. Sun, L. Su, J. Rizo, L. Liu, L.-F. Wang, F.-S. Wang and X. Wang, "Mixed lineage kinase domain-like protein MLKL causes necrotic membrane disruption upon phosphorylation by RIP3", *Molecular cell*, Vol. 54, No. 1, pp. 133–146, 2014.

6. Fu, Z., B. Deng, Y. Liao, L. Shan, F. Yin, Z. Wang, H. Zeng, D. Zuo, Y. Hua and Z. Cai, "The anti-tumor effect of shikonin on osteosarcoma by inducing RIP1 and RIP3 dependent necroptosis", *BMC cancer*, Vol. 13, No. 1, p. 1, 2013.

7. Kedde, M., M. J. Strasser, B. Boldajipour, J. A. O. Vrielink, K. Slanchev, C. le Sage, R. Nagel, P. M. Voorhoeve, J. van Duijse, U. A. Ørom *et al.*, "RNA-binding protein Dnd1 inhibits microRNA access to target mRNA", *Cell*, Vol. 131, No. 7, pp. 1273–1286, 2007.

8. Korkmaz, G., K. A. Tekirdag, D. G. Ozturk, A. Kosar, O. U. Sezerman and D. Gozuacik, "MIR376A is a regulator of starvation-induced autophagy", *PLoS One*, Vol. 8, No. 12, p. e82556, 2013.

9. Huang, C.-C. and Z. Lu, "Community challenges in biomedical text mining over 10 years: success, failure and the future", *Briefings in bioinformatics*, Vol. 17, No. 1, pp. 132–144, 2016.

10. Tan, P., M. Steinbach and V. Kumar, *Introduction to data mining. 1st*, 2005.

11. Lawlor, L. R., "Overlap, similarity, and competition coefficients", *Ecology*, Vol. 61, No. 2, pp. 245–251, 1980.

12. Dice, L. R., "Measures of the amount of ecologic association between species", *Ecology*, Vol. 26, No. 3, pp. 297–302, 1945.

13. Krause, E. F., "Taxicab geometry", *Dover Publ.*, 1987.

14. Ukkonen, E., "Approximate string-matching with q-grams and maximal matches", *Theoretical computer science*, Vol. 92, No. 1, pp. 191–211, 1992.

15. Levenshtein, V. I., "Binary codes capable of correcting deletions, insertions, and reversals", *Soviet physics doklady*, Vol. 10, pp. 707–710, 1966.

16. Smith, T. F. and M. S. Waterman, "Identification of common molecular subsequences", *Journal of molecular biology*, Vol. 147, No. 1, pp. 195–197, 1981.

17. Chvatal, V. and D. Sankoff, "Longest common subsequences of two random sequences", *Journal of Applied Probability*, pp. 306–315, 1975.

18. Needleman, S. B. and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins", *Journal of molecular biology*, Vol. 48, No. 3, pp. 443–453, 1970.

19. Rada, R., H. Mili, E. Bicknell and M. Blettner, "Development and application of a metric on semantic nets", *Systems, Man and Cybernetics, IEEE Transactions on*, Vol. 19, No. 1, pp. 17–30, 1989.

20. Caviedes, J. E. and J. J. Cimino, "Towards the development of a conceptual distance metric for the UMLS", *Journal of biomedical informatics*, Vol. 37, No. 2, pp. 77–85, 2004.

21. Leacock, C. 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.

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

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

24. Lin, D., "An information-theoretic definition of similarity.", *ICML*, Vol. 98, pp. 296–304, 1998.

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

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

27. Patwardhan, S., *Incorporating dictionary and corpus information into a context vector measure of semantic relatedness*, Ph.D. Thesis, University of Minnesota, Duluth, 2003.

28. Miller, G. A., "WordNet: a lexical database for English", *Communications of the ACM*, Vol. 38, No. 11, pp. 39–41, 1995.

29. Shima, H., "WordNet Similarity For Java Library", `http://ws4jdemo.appspot.com/`, 2013, accessed at August 2016.

30. Pedersen, T., S. Patwardhan and J. Michelizzi, "WordNet:: Similarity: measuring the relatedness of concepts", *Demonstration papers at HLT-NAACL 2004*, pp. 38–41, Association for Computational Linguistics, 2004.

31. "Java WordNet Library", `https://sourceforge.net/projects/jwordnet/`, 2013, accessed at August 2016.

32. Bodenreider, O., "The unified medical language system (UMLS): integrating biomedical terminology", *Nucleic acids research*, Vol. 32, No. suppl 1, pp. D267–D270, 2004.

33. "A Python Library for Semantic Similarity", `https://pypi.python.org/pypi/fastsemsim`, accessed at August 2016.

34. Fröhlich, H., N. Speer, A. Poustka and T. Beißbarth, "GOSim–an R-package for computation of information theoretic GO similarities between terms and gene products", *BMC bioinformatics*, Vol. 8, No. 1, p. 166, 2007.

35. Li, J., B. Gong, X. Chen, T. Liu, C. Wu, F. Zhang, C. Li, X. Li, S. Rao and X. Li, "DOSim: An R package for similarity between diseases based on Disease Ontology", *BMC bioinformatics*, Vol. 12, No. 1, p. 1, 2011.

36. McInnes, B. T., T. Pedersen and S. V. Pakhomov, "UMLS-Interface and UMLS-Similarity: open source software for measuring paths and semantic similarity", *AMIA Annual Symposium Proceedings*, Vol. 2009, p. 431, American Medical Informatics Association, 2009.

37. Rumelhart, D. E., G. E. Hinton and R. J. Williams, *Learning internal representations by error propagation*, Tech. rep., DTIC Document, 1985.

38. Rumelhart, D. E., G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors", *Cognitive modeling*, Vol. 5, No. 3, p. 1, 1988.

39. Breiman, L., "Random forests", *Machine learning*, Vol. 45, No. 1, pp. 5–32, 2001.

40. Hastie, T., R. Tibshirani and J. Friedman, "The elements of statistical learning 2nd edition", *New York: Springer*, 2009.

41. Liaw, A., *Documentation for R package random Forest*, Retrieved, 2013.

42. Kenney, J. and E. Keeping, "Linear regression and correlation", *Mathematics of statistics*, Vol. 1, pp. 252–285, 1962.

43. Zhang, T., "Solving large scale linear prediction problems using stochastic gradient descent algorithms", *Proceedings of the twenty-first international conference on Machine learning*, p. 116, ACM, 2004.

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

45. Smola, A. J. *et al.*, "Regression estimation with support vector learning machines", *Master's thesis, Technische Universit at Munchen*, 1996.

46. Gomaa, W. H. and A. A. Fahmy, "A survey of text similarity approaches", *International Journal of Computer Applications*, Vol. 68, No. 13, 2013.

47. Richardson, R., A. Smeaton and J. Murphy, "Using WordNet as a knowledge base for measuring semantic similarity between words", *Technical Report Working Paper CA-1294, School of Computer Applications, Dublin City University*, 1994.

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

49. Li, Y., Z. A. Bandar and D. McLean, "An approach for measuring semantic similarity between words using multiple information sources", *Knowledge and Data Engineering, IEEE Transactions on*, Vol. 15, No. 4, pp. 871–882, 2003.

50. Ahsaee, M. G., M. Naghibzadeh and S. E. Y. Naeini, "Semantic similarity assessment of words using weighted WordNet", *International Journal of Machine Learning and Cybernetics*, Vol. 5, No. 3, pp. 479–490, 2014.

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

52. Meng, L., R. Huang and J. Gu, "A review of semantic similarity measures in wordnet", *International Journal of Hybrid Information Technology*, Vol. 6, No. 1, pp. 1–12, 2013.

53. Lord, P. W., R. D. Stevens, A. Brass and C. A. Goble, "Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation", *Bioinformatics*, Vol. 19, No. 10, pp. 1275–1283, 2003.

54. Pedersen, T., S. V. Pakhomov, S. Patwardhan and C. G. Chute, "Measures of semantic similarity and relatedness in the biomedical domain", *Journal of biomedical informatics*, Vol. 40, No. 3, pp. 288–299, 2007.

55. Batet, M., D. Sánchez, A. Valls and K. Gibert, "Semantic similarity estimation from multiple ontologies", *Applied intelligence*, Vol. 38, No. 1, pp. 29–44, 2013.

56. Mihalcea, R., C. Corley and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity", *AAAI*, Vol. 6, pp. 775–780, 2006.

57. Li, Y., D. McLean, Z. A. Bandar, J. D. O'shea and K. Crockett, "Sentence similarity based on semantic nets and corpus statistics", *Knowledge and Data Engineering, IEEE Transactions on*, Vol. 18, No. 8, pp. 1138–1150, 2006.

58. Agirre, E., M. Diab, D. Cer and A. Gonzalez-Agirre, "Semeval-2012 task 6: A pilot on semantic textual similarity", *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pp. 385–393, Association for Computational Linguistics, 2012.

59. Agirre, E., D. Cer, M. Diab, A. Gonzalez-Agirre and W. Guo, "sem 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity", *In *SEM 2013: The Second Joint Conference on Lexical and Computational Semantics. Association for Computational Linguistics*, Citeseer, 2013.

60. Agirre, E., C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, R. Mihalcea, G. Rigau and J. Wiebe, "Semeval-2014 task 10: Multilingual semantic textual similarity", *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pp. 81–91, 2014.

61. Agirrea, E., C. Baneab, C. Cardiec, D. Cerd, M. Diabe, A. Gonzalez-Agirrea, W. Guof, I. Lopez-Gazpioa, M. Maritxalara, R. Mihalceab *et al.*, "Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability", *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pp. 252–263, 2015.

62. Šarić, F., G. Glavaš, M. Karan, J. Šnajder and B. D. Bašić, "Takelab: Systems for measuring semantic text similarity", *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pp. 441–448, Association for Computational Linguistics, 2012.

63. Han, L., A. Kashyap, T. Finin, J. Mayfield and J. Weese, "UMBC EBIQUITY-CORE: Semantic textual similarity systems", *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*, Vol. 1, pp. 44–52, 2013.

64. Sultan, M. A., S. Bethard and T. Sumner, "Dls@cu: Sentence similarity from word alignment", *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pp. 241–246, 2014.

65. Sultan, M. A., S. Bethard and T. Sumner, "Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence", *Transactions of the Association for Computational Linguistics*, Vol. 2, pp. 219–230, 2014.

66. "Word2Vec Source Code", `https://code.google.com/archive/p/word2vec/`, 2013, accessed at April 2016.

67. Liu, Y., C. Sun, L. Lin, Y. Zhao and X. Wang, "Computing Semantic Text Similarity Using Rich Features", *29th Pacific Asia Conference on Language,Information and Computation*, Vol. 1, pp. 44–52, 2015.

68. "ADW Online Demo", `http://lcl.uniroma1.it/adw/ADWCalculator`, 2013, accessed at April 2016.

69. Vasile Rus, C. M. W. B. N. N. B. M., Mihai Lintean, "SEMILAR Library", `http://www.semanticsimilarity.org/`, 2013, accessed at April 2016.

70. Pilehvar, M. T., D. Jurgens and R. Navigli, "Align, Disambiguate and Walk: A Unified Approach for Measuring Semantic Similarity.", *ACL*, Vol. 1, pp. 1341–1351, 2013.

71. Haveliwala, T. H., "Topic-sensitive pagerank", *Proceedings of the 11th international conference on World Wide Web*, pp. 517–526, ACM, 2002.

72. Rus, V., M. C. Lintean, R. Banjade, N. B. Niraula and D. Stefanescu, "SEMILAR:

The Semantic Similarity Toolkit.", *ACL (Conference System Demonstrations)*, pp. 163–168, Citeseer, 2013.

73. Rus, V., M. Lintean, C. Moldovan, W. Baggett, N. Niraula and B. Morgan, "The SIMILAR corpus: A resource to foster the qualitative understanding of semantic similarity of texts", *Semantic Relations II: Enhancing Resources and Applications, The 8th Language Resources and Evaluation Conference (LREC 2012), May*, pp. 23–25, 2012.

74. Zeno, S., S. Ivens, R. Millard and R. Duvvuri, "The educator's word frequency guide/Touchstone Applied Science Associates", *Inc, Brewster, NY*, 1995.

75. Corley, C. and R. Mihalcea, "Measuring the semantic similarity of texts", *Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment*, pp. 13–18, Association for Computational Linguistics, 2005.

76. "Text Analysis Conference", `http://www.nist.gov/tac/2014/BiomedSumm/`, 2014, accessed at August 2016.

77. "English Stop Words List", `http://www.ranks.nl/stopwords`, 2015, accessed at August 2016.

78. "A Java Library of String Similarity Metrics", `https://github.com/Simmetrics/simmetrics`, 2016, accessed at August 2016.

79. Landauer, T. K. and S. T. Dumais, "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge.", *Psychological review*, Vol. 104, No. 2, p. 211, 1997.

80. David Jurgens, K. S., "The S-Space Package", `https://github.com/fozziethebeat/S-Space`, 2012, accessed at August 2016.

81. "The PMC Open Access Subset of journal literature at U.S. National Institutes of Health's National Library of Medicine", `http://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/`, 2015, accessed at August 2016.

82. Le, Q. V. and T. Mikolov, "Distributed representations of sentences and documents", *arXiv preprint arXiv:1405.4053*, 2014.

83. Dai, A. M., C. Olah and Q. V. Le, "Document embedding with paragraph vectors", *arXiv preprint arXiv:1507.07998*, 2015.

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

85. Aronson, A. R., "Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program.", *Proceedings of the AMIA Symposium*, p. 17, American Medical Informatics Association, 2001.

86. Pearson, K., "Note on regression and inheritance in the case of two parents", *Proceedings of the Royal Society of London*, Vol. 58, pp. 240–242, 1895.

87. Evans, J. D., *Straightforward statistics for the behavioral sciences*, Brooks/Cole, 1996.

88. Tony C. Smith, E. F., "Data Mining Software in Java", `http://www.cs.waikato.ac.nz/ml/weka/`, 2016, accessed at August 2016.

89. Chih-Chung Chang, C.-J. L., "A Library for Support Vector Machines", `https://www.csie.ntu.edu.tw/ cjlin/libsvm/`, 2015, accessed at August 2016.

90. Cohan, A., L. Soldaini and N. Goharian, "Matching Citation Text and Cited Spans in Biomedical Literature: a Search-Oriented Approach", *North American Chapter*

of the Association for Computational Linguistics–Human Language Technologies (NAACL HLT 2015).

91. Erkan, G. and D. R. Radev, "LexRank: Graph-based lexical centrality as salience in text summarization", *Journal of Artificial Intelligence Research*, pp. 457–479, 2004.

APPENDIX A: SYSTEM RESULTS FOR A SUBSET OF THE DATA SET

Table A.1: Ground truth and supervised system scores for selected pairs

Sentence 1	**Sentence 2**	**GT**	**SS**
Changes in miR-146a and miR-146b expression and/or binding have also been implicated in the metastatic and proliferative response associated with the development of papillary thyroid carcinoma (PTC) and cervical cancer, ovarian cancer, breast cancer and pancreatic cancer and prostate cancer.	Additionally, loss of LATS2 stimulated reduplication, an activity comparable to that observed when Cyclin E is overexpressed in the absence of p53	0.2	1.28
Recently, miR-126 was identified as a metastasis suppressing miRNA that is downregulated in relapsing breast cancer, leukemia, and cervical cancer.	Subsequent reports showed that miR-126 targeted the oncogene IRS-1 (insulin receptor substrate-1) in breast cancer cells and miR-126 was downregulated in cervical cancer.	3.2	3.31
Considerable evidence indicates that cancer cells develop dependencies on normal functions of certain genes that can potentially be exploited to improve therapeutic strategies.	In the case of cell response to stress, cyclin D1 can be degraded through its binding to the anaphase-promoting complex and a RXXL sequence located in the NH2-terminal part of the protein.	0	0.34

Table A.1. Ground truth and supervised system scores for selected pairs (cont.)

With respect to LATS2, it has been reported that LATS2 induces G2/M arrest and subsequent apoptotic cell death.	The expression of miR-146a has been found to be up-regulated in papillary thyroid carcinoma, anaplastic thyroid cancer and cervical cancer.	0.2	1.06
Expression of an activated form of Ras proteins can induce senescence in some primary fibroblasts.	The senescent state has been observed to be inducible in certain cultured cells in response to high level expression of genes such as the activated ras oncogene.	3.6	2.94
Three programs, PicTar, miRanda, and TargetScan , were used to predict the targets of miR-21.	The genes that decreased 2-fold or more were further screened for possible miR-372/3 target sites using a local version of the TargetScan algorithm.	2.4	1.84
Human Wts2 is a phosphorylation target of Aurora-A kinase, and this phosphorylation plays a role in regulating centrosomal localization of hWts2.	Similarly to PLK1, Aurora-A activity is required for the enrichment or localisation of multiple centrosomal factors which have roles in maturation, including LATS2 and CDK5RAP2/Cnn.		
miR-223 regulates granulopoiesis by a feedback mechanism and is modulated competitively by the transcription factors nuclear factor I/A (NFI-A) and CCAAT/enhancer binding protein (C/EBP)	There is growing evidence from animal systems that miRNA-regulated transcription factors frequently regulate the transcription of their cognate miRNAs.	1.8	1.90

Table A.1. Ground truth and supervised system scores for selected pairs (cont.)

The oncogenic activity of mutant Kras appears dependent on functional Craf, but not on Braf.	Notably, c-Raf has recently been found essential for development of K-Ras-driven NSCLCs.	1.2	2.78
Oct-4-dependent transcriptional networks have been described regulating self-renewal and pluripotency in human and mouse ES and EC cells and in human mesenchymal cells.	Co-transfection of miRVec-miR-204 and the Renilla-3' UTR plasmid was in HEK293T cells with TransIT-LT1 Transfection Reagent (Mirus).	0	1.02
Consequently miRNAs have been demonstrated to act either as oncogenes (e.g., miR-155, miR-17-5p and miR-21) or tumor suppressors (e.g., miR-34, miR-15a, miR-16-1 and let-7).	Given the extensive involvement of miRNA in physiology, dysregulation of miRNA expression can be associated with cancer pathobiology including oncogenesis], proliferation, epithelial-mesenchymal transition, metastasis, aberrations in metabolism, and angiogenesis, among others.	2.8	1.58
In spite of these caveats, the results of research represent a very important advance in the long-standing fight to conquer lung cancer.	We should consider the data of research as an exciting but early step in the long process of drug discovery.	0.8	1.69

Table A.1. Ground truth and supervised system scores for selected pairs (cont.)

Necrotic death was augmented when caspase activities were compromised by either viral or chemical inhibitors.	Intriguingly, in the presence of caspase inhibitors or following caspase-8 gene ablation, death receptors have also recently been shown to induce necrotic cell death, a process which is dependent on the kinase activity of RIPK1 and RIPK3.	2.8	2.16
This form of necrosis, also termed necroptosis, requires the activity of receptor-interacting protein kinase 1 (RIP1) and its related kinase, RIP3.	TNF-mediated programmed necrosis typically involves the receptor-interacting serine-threonine kinases 1 and 3 (RIP1 and RIP3), as evidenced in human, mouse, and zebrafish cell lines, as well as in a murine sepsis model.	3	2.92
These findings were identical to the pattern of expression seen in human acute lymphoid and myeloid leukemias with MLL rearrangements.	These genes are consistently expressed in leukemias with MLL rearrangements.	3.6	3.12
Since in S. cerevisiae DBF2 was shown to be associated with anaphase and/or telophase progression, we examined whether the deletion of the kinase would also affect cell cycle progression in N. crassa	Taking into consideration the role that DBF2 homologs have been shown to play in cell cycle progression, predominant localization of DBF-2 in N.crassa is expected.	2	2.48

Table A.1. Ground truth and supervised system scores for selected pairs (cont.)

Ironically, Rest has recently been described as both a tumor suppressor and an oncogene.	REST is a transcription factor that represses neuronal genes in non-neural tissues, and plays a prominent tumor suppressor role in epithelial tissues.	3	2.30
The researchers combined available inhibitors selective for two of the pathways regulated by GATA2 to treat mice with Kras-driven NSCLCs.	In PC9 cells, loss of GATA6 and/or HOPX did not alter cell growth, whereas reduction of GATA2 and EGFR inhibited cell viability as previously reported.	1.8	0.87