5-2018

Ontology-based Domain-specific Semantic Similarity Analysis and Applications

Xuebo Song
Clemson University, kuroky.sky@gmail.com

Ontology-based Domain-Specific Semantic Similarity Analysis and Applications

---

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Computer Science

---

by
Xuebo Song
May 2018

Accepted by:
Dr. James Z. Wang, Committee Chair
Dr. Pradip K Srimani
Dr. Jim Martin
Dr. Feng Luo

Abstract

Millions of text data are penetrating into our daily life. These unstructured text data serve as a huge source of information. Efficient organization and analysis of the overwhelming text can filter out irrelevant and redundant information, uncover invaluable knowledge, thus significantly reduce human effort, facilitate knowledge discovery and enhance cognitive abilities. Semantic similarity analysis among text objects is one of the fundamental problems in text mining including document classification/clustering, recommendation, query expansion, information retrieval, relevance feedback, word sense disambiguation, etc. While a combination of common sense and domain knowledge could let a person quickly determine if two objects are similar, the computers understand very little of human thinking. Knowledge resources such as ontologies can greatly capture the semantics of text objects, which enables the numeric representation of both domain knowledge and context information. In this dissertation, we develop a series of techniques to measure the semantic similarity of objects in multiple domains. By utilizing the structured knowledge that has already been established, we explore the domain knowledge from the existing lexical resources and incorporate it into specific applications within different domains. Specifically, we investigate the semantic similarities between gene products using Gene Ontology in biology domain. In text domain, we propose a hybrid representation of text objects (words and documents) based on WordNet which exploits both context and ontology

information to extract meaningful information from the unstructured text to measure the semantic similarity of text documents.

Dedication

I dedicate this humble work to my parents. They have been constant cheerleaders through every academic and personal endeavor in my life. Thanks mom and dad for always believing in me and encouraging me to strive for my dreams. I could not have made this far without their love, support and encouragement during these years.

Acknowledgments

I would like to thank my advisor, Dr. James Z. Wang, for his patience, precious advices and supports throughout my entire Ph.D. period. You have set an example of excellence as a researcher, mentor, instructor and role model.

I would also like to thank my dissertation committee members, Dr. Pradip K Srimani, Dr. Jim Martin and Dr. Feng Luo for all of their guidance through my Ph.D. study. Your discussion, ideas, and feedback have been absolutely invaluable.

I would like to thank Yongtai Liu, Kun he, Runzhen Wang, Rohith Venkatakrishnan, Zongming Yang for being great group mates.

I would like to thank Dr. Zhidian Du, Dr. Lin Li, Dr. Liang Dong, Dr. Yihua Ding, Dr. Yuanyuan Zhang, Dr. Mark Eckert for many useful advices and suggestions.

Finally, I would like to thank the School of Computing and the Graduate School for providing me the wonderful learning experience.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem Statement

We live in a world that's drowning in data. Along with the progress in the area of computer networks, operating systems and the recent big data analysis, numerous technological innovations are driving the dramatic increase of data. Large amounts of data from all aspects of the society are being generated every day. Google receives millions of search queries every minute; Facebook users share more than 2 million pieces of content every day. Digitization of health records has also brought huge opportunities and challenges to the area of medical research.

According to recent estimates, 2.5 quintillion (10^{18}) bytes of data are generated on a daily basis. Large amounts of data across various areas such as education, stocks, sports, news, GPS, need to be stored and analyzed. These data can be shaped into the following structures, video, audio, image, and text. Among those multi-modal data, text data, which is the most intuitive and enriched, becomes an indispensable source of information. Unfortunately, most of this information cannot be used by humans. Either the data is beyond the means of standard analytical methods, or it

is overwhelming to comprehend.

Data mining is considered an automated process of discovering interesting (non-trivial, previously unknown, insightful and potentially useful) information or patterns, as well as descriptive, understandable, and predictive models from (large-scale) data. Even though the massive amount of data may also contain numbers, dates and facts in structured fields, unstructured information is typically text (articles, website text, blog posts, etc.). The presence of unstructured text information makes it more difficult to effectively perform knowledge management activities.

People communicate in many different ways, through speaking and listening, making gestures, or various forms of text. The idea of computers being able to understand ordinary languages and hold conversations with human being has been a dream for centuries. Ever since the 21st century this vision has been starting to look more plausible. Many websites (e.g., Google Translate), now offer automatic translation; Mobile applications can understand speech commands; Search engines can automatically complete or correct your queries and find relevant results that closely match the query terms. However, we are still far from full-fledged machine understanding of natural language. For example, automated translation performs well only when phrases or short sentences are encountered; Speech commands are sensitive to background noises; In search engines, there are still irrelevant search results due to the lack of true understanding of the user's intent.

The fact that computers understand very little of the meaning of human language profoundly limits our ability to give instructions to computers. We are limited in how we can utilize computers to efficiently analyze and process text. Furthermore, it is hard for computers to explain their actions. Semantic similarity is a complex concept which has been widely discussed in the linguistic, philosophical, and information theory communities by computing the similarity between concepts/terms in order to

identify concepts having common "characteristics". Semantic similarity refers to the degree that two concepts are similar (or not). While human do not have a formal definition of similarity between concepts, they can easily determine if two concepts are similar in some way. For example, most people would agree that "pencil" and "pen" are more similar to each other than are "car" and "cellphone". There should be a deep psychological explanation behind human perceptions of similarity and relatedness. Knowledge must be stored in human brain in an efficient and economic fashion. To ascertain the truth of a sentence such as "A canary can fly", consider two possible ways of human organizations of memory. First, people might store the fact that each kind of bird can fly. Then they could retrieve this fact directly to decide the sentence is true. An alternative organization would be to store only the generalization that birds can fly, and to infer that "A canary can fly" from the pre-stored information that a canary is a bird and birds can fly. According to psychology studies [Collins and Quillian, 1969, Olivera, , Quillian, 1969], people tend to store and access knowledge semantically similar in the fashion of the latter organization as it is much more economical in terms of storage space but should require longer retrieval times when such inferences are necessary. While the exact nature of how human access knowledge remains an interesting question, in this dissertation, we consider semantic similarity from a more practical point of view. We try to observe how human beings use this semantic similarity notion in their daily life. A part of human's common sense may include knowing what concepts are similar (or not). For instance, a little child can easily tell that "apple" and "orange" are more similar to each other than "apple" and "desk". Consider the following sentences, "The child is eating an apple" and "The child is eating an orange" is more valid than the sentence "The child is eating a desk". Since "orange" and "apple" is more reasonable to reside in the context of "eat" than "desk". A combination of common sense and domain knowledge about food makes

it clear that "orange" and "apple" are both food and hence edible while "desk" is not. From this perspective, we can say "orange" and "apple" are more similar to each other than to the word "desk". The combination of domain knowledge and common sense also indicate that the "orange" being referring to is one kind of fruit, and is not associated with the well known color. These are kinds of problems that human can solve quickly, without lots of conscious thought, based on a combination of real world knowledge and common sense.

The study of semantic similarity measurements has long been an integral part of information retrieval and natural language processing. Semantic similarity between entities changes over time and across domain [Cilibrasi and Vitanyi, 2007]. For instance, "apple" is frequently associated with "computers" on the Web. However, this sense of "apple" is not listed in most general thesaurus or dictionaries. Users who search for "apple" might be interested in this sense of "apple" rather than the "apple" as a fruit. New words are constantly being generated as well as new senses are assigned to existing words. Manually maintaining the thesaurus or dictionaries to capture these new words and senses is too expensive if not impossible.

We may wonder if it is possible to develop computer programs to make the same kind of judgment as human beings? While the answer is yet to be found, a more reasonable question might be: *Can we automate and quantify semantic similarity, so as to correspond with human judgment?* The answer to this question, based on previous research and this dissertation, is a sound yes. Despite the mystic issues above, it still reasonable to say that human beings are largely in agreement on the semantic similarity of concepts.

However, this could be challenging as there are a wide variety of ways that concepts that can be related, and it might require a certain amount of specialized knowledge to explore the inner relationships. For example, at first look the automotive

sense of "tire" and the "eraser" from stationery may not seem to be similar or related at all. However, if one is aware that both are made of rubber, then they may be more similar or related than first realized. In addition, humans are not always in complete agreement on similarity or relatedness judgments, since these can be highly affected by personal experiences. For example, a particular person may consider "tire" and "eraser" to be highly similar just because that person has a eraser of the shape of a car.

The differences between *words, concepts* and *word senses* are subtle. Concepts are real-world objects, ideas that are represented in text or speech by words. For instance, the concept of a car can be represented by the word "car". It may also be represented by the word "automobile" or "motorcar". Hence, the same concept may be represented by different words. We call these word referring to the same concepts *synonyms*, or the relationship between these words *synonymy*. In addition, a concept need not be a solid object. It could be an abstract thing, like motion. Each such concept can also have a number of words that represent it. Nevertheless, a single word may express a number of concepts. For instance, the word "table" could mean a tabular array or a piece of furniture. The different meanings of a word are known as *word senses*. The phenomenon of the coexistence of many possible meanings for a word are called *polysemy*. A word may, therefore, correspond to a number of different concepts, while a word sense corresponds only to a single concept. Considering this equivalence of word senses and concepts, we use the term concepts and word sense interchangeably in this dissertation.

However, in certain scenarios (e.g., short message analysis), information contained in the text may not be sufficient in deriving semantics. Hence, incorporating information from knowledge resources such as ontologies becomes an important part in semantic similarity analysis. The fact that ontologies can provide an efficient way

to reduce the amount of information overload by encoding the structure of a specific domain and offering easier access to the information could shed lights on semantic similarity analysis of text objects.

1.2 Dissertation Summary

In this dissertation, we develop a series of techniques to measure the semantic similarity of objects in multiple domains. By utilizing the structured knowledge that has already been established, we explore the domain knowledge from the existing lexical resources and incorporate it into specific applications within different domains. Specifically, we investigate the semantic similarities between gene products using Gene Ontology, the semantic similarities among words (concepts) and those of documents with the help of WordNet. We develop similarity metrics specifically designed for gene products. We first develop semantic similarity metrics between terms that are used to annotate gene products. We further extend to compute the functional similarities between gene products based on their annotations. Unlike gene products that can be annotated by a small portion of the entire ontology by human experts [Consortium, 2013], the large amount of information within the text makes it too expensive to annotate words by human labor, if possible at all. The fact that the same word can refer to different meanings in different contexts only aggravates this situation. One of the most important observations is that the semantics of words in text documents could be highly related to surrounding text or similar words. Moreover, the irrelevant information (noises) in text documents might highly mislead the semantics of words. Nevertheless, people usually express the same meaning using similar or related words widely spread across the text. Although these similar or related words might convey important information about the text, the relations be-

tween them cannot be identified solely based on context. It is important that we can incorporate these prior information (pre-established knowledge) into text analysis. We propose a hybrid representation of words which combines information from context and ontology. The new representation can efficiently utilize knowledge outside the text and enrich text content in semantic similarity measurements. It would greatly benefit natural language processing tasks and applications (E.g., document classification and clustering).

1.3 Research Contributions

In this dissertation, we study the semantic similarity metrics between terms in different domains using specified ontologies. We first investigate the semantic similarity of gene products using Gene Ontology. We also explore an ontology-assisted approach to generate word representations that can capture more semantics from both the background knowledge using WordNet and the context information using neural networks. We further extend the word representations to document representations and apply a simple approach to calculate document similarities.

Aggregate Information Content (AIC): we propose a novel and efficient method to measure the semantic similarity of GO terms. The proposed method addresses the limitations in existing GO term similarity measurement techniques; it computes the semantic content of a GO term by considering the information content of all of its ancestor terms in the graph. The aggregate information content (AIC) of all ancestor terms of a GO term implicitly reflects the GO term's location in the GO graph and also represents how human beings use this GO term and all its ancestor terms to annotate genes. We show that semantic similarity of GO terms obtained by our method closely matches the human perception. Extensive experimental studies

show that this novel method also outperforms all existing methods in terms of the correlation with gene expression data. We have developed Web services for measuring semantic similarity of GO terms and functional similarity of genes using the proposed AIC method and other popular methods. These Web services are available at `http://bioinformatics.clemson.edu/G-SESAME`.

Hybrid Word Embeddings (HWE): we propose a novel text representation method, HWE, which combines semantic information obtained from WordNet and context information extracted from text documents to provide concise and accurate representations of text documents. Compared to existing word embeddings based approaches, such as Doc2Vec and Word2Vec, the proposed HWE method can improve the efficiency of deriving word semantics from text by taking advantage of the word semantic relationships extracted from WordNet. Experimental study on classification of documents shows that HWE outperforms the state-of-the-art methods, including Doc2Vec and Word2Vec, in terms of classification accuracy, recall, precision, etc. Unlike traditional document representations which need very large corpus as input to create sparse representations and project them into a lower dimensional dense vector space, including Doc2Vec [Le and Mikolov, 2014] and Word2Vec [Mikolov et al., 2013b]. The proposed HWE model can use much less data with the help of existing knowledge resources like WordNet. Moreover, in the scenario where a document contains new words that have not appeared in the training set, both Word2Vec and Doc2Vec fail to capture the word semantics without re-training the entire corpus. If the new words can be found in WordNet, the proposed HWE model is more flexible to derive the semantics of new words as it can utilize WordNet to attribute new words to related concepts that have already been trained. In the case where new words are not recorded by WordNet, HWE can also achieve better results by utilizing the global information than Word2Vec or Doc2Vec by re-training the entire corpus. Nowadays,

Words are endowed with new semantic meanings rapidly in the light of large volumes of news feed, tweets, blogs, etc. This rapid semantic change or enrichment can not be well captured by a single thesaurus or ontology. The proposed HWE can alleviate this problem by taking advantage of both context information as well as the pre-established structured knowledge in ontologies.

1.4 Dissertation Organization

The rest of this dissertation is organized as follows, the background information of the ontology, ontology-based semantic similarity metrics and word/document representation techniques are presented in chapter 2. In chapter 3, Aggregate Information Content (AIC) is introduced to the compute the semantic similarity of GO terms and gene products accurately and efficiently in biology domain using Gene Ontology. A hybrid word representation model incorporating domain knowledge extracted from WordNet to calculate text similarity is introduced in Chapter 4. Combining the domain knowledge with the context in word representations are expected to alleviate the situation where synonmy and polysemy heavily reduce NLP system performances. Conclusions and future works can be found in Chapter 5.

Chapter 2

Ontology, Word/Term similarity and Word Representation

An efficient numerical measurement of semantic similarity is critical to many applications of natural language processing (NLP) systems, such as information retrieval schemes (identify an optimal match between search query terms and documents) [Budanitsky and Hirst, 2006, Kobayashi and Takeda, 2000], thesauri generation [Curran, 2002], information extraction [Atkinson et al., 2009, Stevenson and Greenwood, 2005], word sense disambiguation [Patwardhan, 2003, Simov et al., 2016], text classification and clustering [Lin et al., 2014, Kogan et al., 2005, Dhillon et al., 2003, Chim and Deng, 2008, Sun and Lim, 2001, Lodhi et al., 2002] and bioinformatics [Nguyen and Al-Mubaid, 2006, Song et al., 2014, Song et al., 2013, Wang et al., 2007]. A number of measures of semantic similarity have been developed and evaluated by researchers, these representation can be categorized as ontology-based methods [Resnik, 1999, Jiang and Conrath, 1997, Lin, 1998, Wang et al., 2007, Song et al., 2014, Snchez et al., 2012, Cimiano, 2006, Taieb et al., 2014, Bollegala et al., 2007], web-based measures [Cilibrasi and Vitanyi, 2007, Bollegala et al., 2007] and word

representation-based approaches [Mikolov et al., 2013b, Le and Mikolov, 2014, LANDAUER and DUMAIS, 1997, Salton and McGill, 1986]. In this chapter, we will review and discuss some of the most significant ones.

2.1 Domain Ontology

According to [Horrocks, 2008], ontology, in its original philosophical sense, is a branch of metaphysics focusing on the study of existence whose objective is to study the structure of the world by determining what entities and types of entities exist. In computer science, an ontology can be considered an engineering artifact, usually a model of (part of) the world; it introduces vocabulary describing a variety of aspects of the domain being modeled and provides an explicit specification of the intended meaning of that vocabulary.

An ontology can be seen as an information model that explicitly describes various entities and concepts in a domain along with their properties and relations. These abstract description systems for domain-specific knowledge receive more and more attention in text mining and bioinformatics studies. With the growing demand of knowledge organization and information reuse, ontology as a semantic and knowledge model has aroused many concern of researchers, and widely applied in many areas of computer science such as knowledge engineering, digital libraries, information retrieval and semantics web. One of the most important tasks involving ontology analysis is to quantitatively measure the relationships between words (terms) in the ontologies, necessitating the computational methods of ontology-based semantic similarity between words (terms).

From the view point of knowledge sharing, the ontology can be conceptualized as an explicit description of the objective existence. It abstracts certain application

field of the real world into a set of concepts and relationships of concepts. An ontology can be viewed as a semantic graph, an important and natural approach to representing real-world knowledge. Domain ontology, describing a set of representational primitives with which to model a domain knowledge of discourse, provides a common and unambiguous understanding of a domain. The benefits of ontology have enabled researchers to incorporate more semantics into traditional information retrieval and NLP techniques. In addition, the use of ontologies to represent concepts and relations among concepts can greatly capture more semantics and may provide insights in exploiting knowledge resources to measure semantic similarities.

2.2 Ontology-Based Word/Term Similarity

Standard alphabetical procedures for organizing lexical information put words/terms together that are spelled alike and are not related by meaning. Knowledge about concepts is huge, while a look-up in the traditional dictionary could be tedious and time-consuming, it must be stored in human brain in an efficient and economic fashion. According to psychology studies, human tends to store and access knowledge semantically similar. Knowledge about concepts are computed "on the fly" via access to general concepts. E.g, we know that "canaries fly" because "birds fly" and "canaries are a kind of bird". This associative process might indicate a hierarchical structure animal $\rightarrow$ bird $\rightarrow$ canary. In general, it is believed that knowledge is stored at the highest possible position and inherited by lower (more specific) concepts rather than being randomly stored.

The idea inspired ontologies such as WordNet and Gene Ontology. These lexicons can be represented as semantic networks and words/terms are interlinked by meaning. Ontology-based similarity analysis tries to identify the degree of sim-

ilarity between words/terms using information derived from such ontology graphs. Ontology-based semantic similarity metrics can be roughly categorized as edge-counting, information content-based, feature-based, gloss-based and hybrid measures.

- Edge-counting measures try to measure similarity based on the number of semantic links and the minimum path length between two concepts present in a given ontology [Li et al., 2003, Rada et al., 1989, Leacock and Chodorow, 1998, Wu and Palmer, 1994].

- Information content-based methods compute the similarity between concepts as a function of the information content (IC) that both concepts have in common in a given ontology [Sebti and Barfroush, 2008, Hadj Taieb et al., 2013, Yuan et al., 2013, Zhou et al., 2008a, HadjTaieb et al., 2014, Resnik, 1999, Lin, 1998, Jiang and Conrath, 1997].

- Feature-based approaches estimate similarity according to the weighted sum of the number of common and non-common features. In addition to concept descriptions, both taxonomic and non-taxonomic information are also considered [Snchez and Batet, 2011, Petrakis et al., 2006, Rodríguez and Egenhofer, 2003]. However, theses methods usually reply on non-taxonomic features that are rarely found in ontologies [Ding et al., 2004].

- Gloss-based methods exploit the definitions provided by the ontology in order to quantify the overlaps between the glosses of two concepts with their semantic neighbors [Banerjee and Pedersen, 2003, Budanitsky and Hirst, 2006, Patwardhan, 2006].

- Hybrid measures combine methods from different methods in order to accumulate the advantages of these measures [Zhou et al., 2008b, Wang et al., 2007].

The following is an overview of the four most representative methods for ontology-based semantic similarity measure: Method A by Resnik [Resnik, 1999], Method B by Lin [Lin, 1998], Method C by Jiang and Conrath [Jiang and Conrath, 1997], and Method D by Wang et al [Wang et al., 2007].

Method A: The frequency of an ontology term is recursively defined as,

$$freq(t) = annotation(t) + \sum_{i \in child(t)} freq(i) \tag{2.1}$$

where $annotation(t)$ is the number of gene products annotated with term t in the database. $child(t)$ is the set of children of term t. For each term t, $p(t)$ denotes the probability that term t occurs in the database,

$$p(t) = freq(t)/freq(root) \tag{2.2}$$

Information Content(IC) of term t is defined as

$$IC(t) = -\log p(t) \tag{2.3}$$

Method A uses ***Maximum Information Contained in Ancestors (MICA)*** of two terms to measure the semantic similarity between them.

$$sim(a, b) = \max_{c \in P(a,b)} IC(c) \tag{2.4}$$

where $P(a, b)$ denotes the set of common ancestor terms of term a and term b in the ontology graph. Based on the definition of IC in Method A (Equations 2.1, 2.2, 2.3), MICA often happens to be the IC value of the ***Least Common Ancestor (LCA)***

of terms a and b.

The principal limitation of method A derives from the fact that it considers only MICA of two terms while ignoring the distances of the two terms to their LCA and the semantic contribution of other ancestor terms. For example, terms a and b have the same LCA with terms c and b in the partial graph shown in Figure 2.1. Using method A, the semantic similarity between term a and b would be equal to the semantic similarity between term c and d, inconsistent with human perception, which suggests that the semantic similarity between term c and d should be less than the one between term a and b.

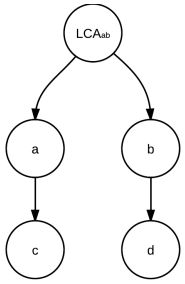

Figure 2.1: Terms at different ontology levels sharing the same LCA

Method B: It is based on the ratio between IC values of two terms and that of their MICA; the semantic similarity between two terms a and b is defined as,

$$sim(a,b) = \frac{2 * max_{c \in P(a,b)} IC(c)}{IC(a) + IC(b)} \tag{2.5}$$

Method C: It introduces the concept of term distance into the semantic similarity calculation. The intuition is that two terms closer in the graph should be more similar than two terms farther in the graph. The distance between two terms a and b is defined as

$$Dis(a,b) = IC(a) + IC(b) - 2 * \max_{c \in P(a,b)} IC(c) \tag{2.6}$$

The semantic similarity of two terms a and b are then defined as

$$sim(a,b) = \frac{1}{1 + Dis(a,b)} \tag{2.7}$$

Note: Methods B and C ameliorated the principal limitation of Method A by implicitly considering the graph distance of the two terms in the semantic similarity measure. Consider the example in Figure 2.1; $sim(c,d)$ should be less than $sim(a,b)$ according to human perception because the graph distance between c and d is greater than the graph distance between a and b. Since term a is a parent of term c, we have $freq(a) > freq(c)$ and $p(a) > p(c)$ (Equations 2.1 and 2.2). According to the definition of IC in Equation 2.3, we have $IC(c) > IC(a)$. Similarly, we have $IC(d) > IC(b)$. Therefore, the semantic similarity values obtained by both methods B and C are consistent with human perception in this aspect.

However, it is possible that a term has multiple parent terms with different semantic relations (Gene Ontology); using MICA alone does not account for multiple parents. Also, two terms at a higher level (more general terms) of the graph should be, as is perceived by humans, semantically more dissimilar than two terms with the same graph distance at a lower level (more specific terms). Since neither methods B nor C factor in the specialization level of the LCA of the two terms in their semantic similarity measure, the semantic similarity values obtained by these two methods may still be inconsistent with human perception.

Method D: Method D attempts to address the shortcomings of other methods by aggregating the semantic contributions of ancestor terms in the graph. The S-value of term t related to term a (where term t is an ancestor of term a, including term a

itself) is defined as,

$$S_a(t) = \begin{cases} 1 & \text{if } t = a \\ \max\{w_e * S_a(t') |\ t' \in \text{children of } t\} & \text{if } t \neq a \end{cases} \tag{2.8}$$

where w_e is the ***semantic contribution factor*** of an edge (weight of the edge in the graph). Then the ***semantic value (SV)*** of a term a is defined as,

$$SV(a) = \sum_{t \in T_a} S_a(t) \tag{2.9}$$

where T_a is the set of terms in DAG_a (Directed Acyclic Graph consisting all ancestors of the term a, including term a). Finally, the semantic similarity between two terms a, b is defined as,

$$sim(a, b) = \frac{\sum_{t \in T_a \cap T_b} (S_a(t) + S_b(t))}{SV(a) + SV(b)} \tag{2.10}$$

where $S_a(t)$ is the S-value of term t related to term a and $S_b(t)$ is the S-value of term t related to term b. While this method combines both the semantic and the topological information of terms to address weaknesses of methods A, B and C, it still suffers from two disadvantages. First, it needs to use semantic contribution factor values (weight) empirically obtained from gene classification to calculate the semantic values of terms. Using semantic contribution factors obtained from the classification of genes from certain species may not be suitable for measuring the functional similarity of genes in other species. Second, some biomedical studies need to obtain the similarity matrix for a large group of terms or genes. Dynamically calculating the semantic values of terms is time consuming and may result in a long user response time, which will be introduced in Section 3 as our previous research.

2.3 Web-Based Word Similarity

The world-wide-web is the largest databases on earth, and the context information entered by millions of independent users averages out to provide automatic semantics of useful quality. In another words, the large amount of information about very conceivable topic makes it likely that extremes will be cancel out and the majority or average is meaningful to some extent. One of the most representative methods in this category is the Google similarity distance [Cilibrasi and Vitanyi, 2007]. It is established on the page count of a given query, which is the number of pages that contain the query words in the google web search. Page count for the query $PANDQ$ can be considered as a global measure of the co-occurrence of word P and Q. For example, the page count of the query "computer" AND "apple" in Google is 9,670,000, whereas the page count of the query "orange" AND "computer" is 3,220,000. This might indicate that "apple" is more semantically similar to "computer" than is "orange". The *Normalized Google Similarity Distance (NGD)* is defined as,

$$NGD(x,y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x,y)}{\log N - \min\{\log f(x), \log f(y)\}} \tag{2.11}$$

where $f(x)$ denotes the number of pages containing x, and $f(x,y)$ denotes the number of pages containing both x and y, as reported by Google.

The two biggest advantages of [Cilibrasi and Vitanyi, 2007]'s method are the computation efficiency and domain coverage. First, the only information used in the calculation of the NGD is the page counts that can be returned by Google in less than one second. Second, the similarity measurements do not reply on domains. Using NGD, we can calculate the semantic similarity of two arbitrary words, even if they may not be related at all in human perception. The performance of Google Similarity

Distance along with other web-based methods [Bollegala et al., 2007] highly relies on the accuracy of the returned counts. Furthermore, Google estimates the number of hits based on samples, and the number of indexed web pages changes rapidly.

2.4 Word Representation

The advantage of semantic similarity measures based on the ontology structure is that they only use an ontology as background knowledge (i.e., no corpus with domain data is needed). However, the main problem is that they heavily depend on the degree of completeness, homogeneity and coverage of the terms and semantic links represented in the ontology [Cimiano, 2006]. While knowledge-based semantic similarity between words can model the relatedness between words/concepts, similarity between text is yet to explore. Text is composed of words or phrases and the relation between text is much more complex than the relation between words. Foremost, the context plays a very important role when determining the true semantics of the text. For example, consider the sentence "The kids love baseball, so I got him a bat and glove." The combination of common sense and domain knowledge about sports makes it clear that the "bat" is the one that is used to hit balls rather than the well know mammal. If the word "bat" is associated with the word "baseball" in the sentence, we could easily that the "bat" is within the sport domain. Hence, to fully exploit the text similarity, efficient representation of text or words that can account for context information could reduce the ambiguity when determining the true semantics of the text.

Data representation is a fundamental task in machine learning and data mining. For a long time, data representation is performed by feature engineering using domain knowledge with the help of experts for designing better features for specific

tasks, which is both difficult and computationally expensive for training. Recently, the rapid development of deep learning has brought new inspiration.

The first and arguably most important common problem across all NLP tasks is how we represent words. A good representation of words can benefits the exploitation of relations between words, especially the similarity and difference. A word representation is a mathematical object associated with each word, often a vector. Each dimension's value corresponds to a feature and might even have a semantic or grammatical interpretation. Among the 13 million tokens for the English language, one might be related to another. It might be easier to analyze the relatedness between words if we encode word tokens into some vector that represents a point in the word space. There might actually exists some N-dimensional space ($N \ll 13$ million) that is sufficient to encode all semantics of human language. Each dimension would encode some meaning that we transfer. For example, semantic dimensions may indicate tense (past vs. present vs. future), and count (singular vs. plural) etc.

Word representation is central to natural language processing. Many methods of deriving word representations were explored in the NLP community. On one end of the spectrum, words are grouped into clusters based on their contexts. On the other end, words are represented as very high dimensional but sparse vectors in which each dimension is a measure of the association between the word and a particular context. The followings are the most representative techniques of word representation methods.

2.4.1 One-Hot Vector

The most simple representation of words is the **one-hot vector** [Harris and Harris, 2012]: Represent every word as an $\mathbb{R}^{|V|\times 1}$ vector with all 0s and one 1 at the index of that word in the sorted vocabulary as shown in Figure 2.2, where $|V|$

is the size of the vocabulary. while one-hot vectors could be stored efficiently, their

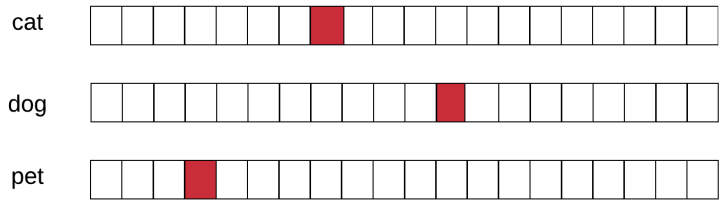

Figure 2.2: An illustration of one-hot vectors of cat, dog and pet. The only non-zero entry in each vector is the index of the word in the vocabulary.

main problem is that they don't capture any information about the relation (such as similarity) between words.

2.4.2 Word-Document Matrix

The word-document matrix is the representation of documents that consists of rows of documents and columns of words (Figure 2.3), which builds on the assumption that words that are related will often appear in the same documents. For instance, "fumble", "touchdown", "score", "intercept", etc. are probably likely to appear together, while "fishing", "papers", "speeding", and "city" would probably not appear together. Each value in the word-document matrix consists of the weight for a specific term in a specific document. Multiple methods are available to assign weights for each element in the matrix such as using binary value indicating the presence of the word in the document or term frequencies and inverse document frequencies (TF-IDF)[Salton and McGill, 1986] in Equation 2.12.

$$tf - idf_{t,d} \;=\; tf_{t,d} \times idf_t \;=\; tf_{t,d} \times log(\frac{N}{df_t}) \tag{2.12}$$

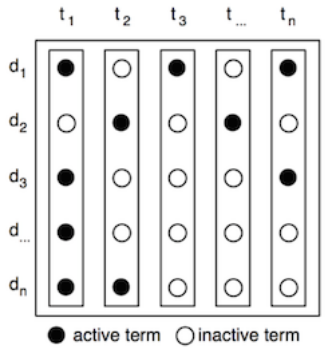

Figure 2.3: Word-Document matrix representation

where $tf_{t,d}$ is the term frequency of t in document d, idf_t is the inverse document frequency of the term t, df_t is the document frequency of term t, N is the total number of documents. These weighting schemes could be used to retrieve relevant document from the document collection for a given query. However, the Word-Document Matrix is a very large matrix and it scales with the number of documents which could lead to disasters when the number of documents is huge.

We can also perceive each word-document matrix row as a vector of word weights. This representation of a set of documents as vectors in a common vector space is also known as the *Vector Space Model*. In most scenarios, due to the large amount of words in the thesaurus, only a minor part of the words in Figure 2.3 are used to represent each document which indicates that each document vector mostly contains zeros, aka., sparse vectors.

The advantage of representing each document as a vector is that we would have the ability to use geometric methods to encode the properties of each document. The documents that share the same set of terms tend to be very close to each other when plotting the document vectors in an n dimensional space. In Figure 2.4, three document vectors are plotted. The closer two document vectors are, the more similar the content of the two documents is to each other. The similarity between two

documents in VSM can be computed by taking the cosine of the angle between the two document vectors. The similarity of document A and B can be taken as,

$$cos(\theta) = \frac{A \cdot B}{|A||B|} \tag{2.13}$$

The range of the above cosine similarity is -1 to 1, where 1 indicates that two document vectors are exactly the same and 0 means that the two vectors are independent and -1 implies that the two vectors are exactly opposite.

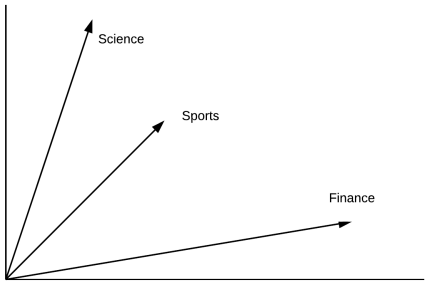

Figure 2.4: Document vectors in Vector Space Model

2.4.3 Latent Semantic Indexing

Latent Semantic Indexing (LSI), sometimes referred to as Latent Semantic Analysis (LSA), is an indexing and retrieval method to identify relationships among terms and concepts in an unstructured collection of text. The concept of LSI was patented in 1988 by a group of researchers at Bell Communications Research [LANDAUER and DUMAIS, 1997]. LSI attempts to analyze the conceptual content of the documents by a statistical analysis of the latent structures of the documents. It takes a vector space representation of documents based on term frequencies as a starting point and applies a dimension reduction operation on the corresponding term/document matrix using the singular value decomposition (SVD) algorithm

[Golub and Reinsch, 1970]. With the rank reduction of the original word-document matrix, similarities among documents and queries can be more reliably estimated in the reduced space than in the original representation. This is because that share frequently co-occurring terms will tend to have a similar representation in the reduced space representation. LSI is commonly used in information retrieval tasks such as web retrieval, document indexing and feature identification. The idea behind LSI is that the original thesaurus consists of a multitude of terms that have the same meaning in the underlying latent structure when the redundant dimensions are removed by LSI.

2.4.4 Language Models

All the above methods require computing and storing global information on some huge data sets. Instead, a model can be created such that it learns one iteration at a time without the knowledge of global information. The probabilistic model can be set up so that it takes one training example at a time. Each time it tries to learn just a little bit of information for the unknown parameters of the model. The model can be evaluated at every iteration based on the input, output, and the label of the output in the model, to follow an update rule that penalizing the model parameters that caused the error. A statistical language model is a probability distribution over sequences of words, i.e., it assigns a probability to a sequence of words. The model aims to give high probabilities to valid sentences, both semantically and syntactically, such as "The food in this restaurant is amazing.". On the other side, sentences that do not make sense should have low probabilities. This model can be formularized as,

$$p(w^{(1)}, w^{(2)}, w^{(3)}, \ldots, w^{(m)}) = \prod_{i=1}^{m} p(w^{(i)} | w^{(1)}, \ldots, w^{(i-1)}) \tag{2.14}$$

The overwhelming number of parameters involved in estimating the probability makes it unrealistic as we will never see enough data, which is called the data sparsity problem (phenomenon of not observing enough data in a corpus to model language accurately). The model is simplified as n-gram under Markov Assumption that the probability of the sequence consisting of m words depends on a word in the sequence and the preceding n-1 words,

$$p(w^{(1)}, w^{(2)}, w^{(3)}, \ldots, w^{(m)}) = \prod_{i=1}^{n} p(w^{(i)}|w^{(i-n+1)}, \ldots, w^{(i-1)}) \tag{2.15}$$

Trigram (n=3), bigram (n=2), and unigram (n=1) are typical n-gram models. n-gram models calculate the probability of each pair of words by counting their co-occurrences. Although these simplified models could extensively reduce the computational cost, the vast amount of calculations of the occurrences in a big corpus necessitates the pre-computation of all the probabilities with huge space to store. Nevertheless, concerning only preceding n words rather than evaluating a whole sentence will lose information especially for the one that has long-distance dependencies. For instance, the sentence "The boy playing in the Olympic park looks very happy." cannot be captured by the trigram model or bigram model and higher-order models (four-grams and above) suffer so much from data sparseness that they become unusable. As a matter of fact, n-gram models only considers pairs of neighboring words rather than evaluating a whole sentence. Language models can be put into a more general form,

$$p(w^{(1)}, w^{(2)}, w^{(3)}, \ldots, w^{(m)}) = \prod_{i=1}^{m} p(w_i|Context_i) \tag{2.16}$$

$Context_i$ represents the context of the word w_i, i.e., the surrounding words of w_i. Specifically, when $Context(w_i)$ is empty, $p(w_i|Context_i)$ is reduced to $p(w_i)$, the unigram model. Taking the *log*, the objective function is regarded as,

$$p(w^{(1)}, w^{(2)}, w^{(3)}, \ldots, w^{(m)}) = \prod_{i=1}^{m} \log p(w_i|Context_i) \tag{2.17}$$

Instead of extracting the occurrences of words out of the corpus, we can directly compute the probability of $p(w_i|Context_i)$ through a function, described as,

$$p(w_i|Context_i|) = F(w_i, Context_i, \theta) \tag{2.18}$$

where θ is the unknown parameter set. Once an optimized parameter set θ^* is obtained by optimizing Equation 2.17, $p(w_i|Context_i)$ can be directly computed through $F(w_i, Context_i, \theta^*)$. Compared to n-gram models, there is no need to pre-compute all the probabilities and the number of parameters could be much lesser than those in n-gram models.

2.4.5 Word Embeddings

The term word embeddings was originally introduce by [Bengio et al., 2003] who trained them in a neural language model. However, [Collobert and Weston, 2008] was arguably the first to demonstrate the power of word embeddings where they establish word embeddings as an effective tool in tasks such as multitask learning and semi-supervised learning from unlabeled text. [Mikolov et al., 2013b] brought word embeddings to the fore with the creation of Word2Vec, a toolkit enabling the training and use of word embeddings.

Word embeddings are a type of word representation that are learned to allow

words with similar meanings to have a similar representation. They are distributed word representations where words are mapped into dense, low-dimensional and real-valued vectors that can capture the semantic and syntactic properties. Most recently, words embeddings are derived by various training methods inspired from neural network language modeling [Mikolov et al., 2013b, Le and Mikolov, 2014, Pennington et al., 2014, Collobert and Weston, 2008]. These embeddings, referred to as "neural embeddings", are used abundantly by machine learning algorithms across a variety of tasks such as text classification and clustering. The fact that they do not require pricey annotation can greatly benefit NLP tasks. Using dense and low-dimensional vectors can also greatly reduce the computational costs as the majority of neural network toolkits do not work well with very high-dimensional, sparse vectors. The basic idea is to train neural network models to get optimized $F(w_i, Context_i, \theta^*)$ in Equation 2.17 and store the same contextual information in a low-dimensional vector; each vector is now represented by a D-dimensional vector, where D is a relatively small number.

Neural network based language model generating word embeddings such as Word2Vec [Mikolov et al., 2013b] can reduce the dimension of the word vectors. However, it still fails to capture the long-distance semantic relations as the only relationships they capture are from a context window, which usually contains 5-10 words. Moreover, the training corpus these models reply on can be extremely large in order to achieve good performance. As Word2Vec only focuses on the context information of words, it cannot be applied to tasks such as short message analysis. The purpose and usefulness of Word2Vec is to group the vectors of similar words together in vector space and detect similarities mathematically. Word2Vec creates distributed numerical representations of word features (the context of individual words) automatically without human intervention.

Given enough data, usage and contexts, Word2Vec can make highly accurate guesses about a words meaning based on past appearances. Those guesses can be used to establish the association with other words (e.g. man is to boy what woman is to girl), or cluster documents. The document clusters can form the basis of search, sentiment analysis and recommendations in such diverse fields as scientific research, legal discovery, e-commerce and customer relationship management. The output of the Word2Vec neural net is a vocabulary in which each item has a vector attached to it, which can be fed into a deep-learning net or simply queried to detect relationships between words.

The distributional hypothesis states that words occurring in similar contexts (with the same neighboring words) tend to have similar meanings (e.g. *hamburger*, *sandwich*) will both appear next to (*eat*, *delicious*, *restaurant*). First of all, despite the low dimensions of word embeddings, the information regarding word similarity is kept. Similar words still have similar vectors. Second, they are compact. Any operation on these vectors (e.g. computing similarities) is efficient and fast. We can also use Word2Vec to solve analogy questions: a is to b as c is to d; given a, b and c, find the missing word d. This is done by addition and subtraction. An example would be "$king + woman - man = queen$".

Using ontologies as background knowledge can help capture the structured domain knowledge in semantic similarity measurements. However, the fact that ontologies heavily depend on the degree of completeness, homogeneity and coverage of the terms and semantic links represented in ontologies greatly affects the flexibility and quality of the semantic similarity measurements. Word semantics in documents cannot be solely identified from ontologies nor can be fully captured by the surrounding context. It is important that we can incorporate background knowledge from ontologies into semantic measurements in different domains depending on the appli-

cations. In the next two sections, we further explore the ontology-based semantic similarity metrics in multiple domains. Specifically, we first discuss how to use the Gene Ontology to explore pre-established structural information as well corpus statistics to efficiently measure the functional semantic similarity of gene products. After which, we capture context information with the help of WordNet, a more general ontology that covers terms in our daily life to calculate the semantic similarity between text documents.

Chapter 3

Semantic Similarity Analysis of Gene Products

Ontology plays a very important role for many knowledge-intensive applications to which they provide source of precisely defined terms. Traditional text mining approaches either focus on term frequencies within documents and across document collections (statistical approaches) or on the structural information (ontology-based approaches). The freedom from biases can be an advantage, but at the cost of ignoring potentially valuable knowledge. Statistical approaches may need extra knowledge from the structured ontology to improve performance of text mining systems, while ontology-based methods may need information from the corpus to adapt to specific applications. A hybrid approach integrating both information sources (corpus and ontology knowledge) could be superior to either method.

Gene Ontology (GO) describes the attributes of genes and gene products (either RNA or protein, resulting from expression of a gene) using a structured and controlled vocabulary. GO consists of three ontologies: biological process (BP), cellular component (CC) and molecular function (MF), each of which is modeled as a

directed acyclic graph. In recent past, many biomedical databases, such as Model Organism Databases (MODs) [Stein et al., 2002], UniProt [Consortium, 2008], SwissProt [Kriventseva et al., 2001], have been annotated by GO terms to help researchers understand the semantic meanings of biomedical entities. With such a large diverse biomedical data set annotated by GO terms, computing functional or structural similarity of biomedical entities has become a very important research topic. Many researchers have tried to measure the functional similarity of genes or proteins based on their GO annotations [Xu et al., 2008, Wang et al., 2007, Wang et al., 2004, Sevilla et al., 2005, Schlicker et al., 2006, Cheng et al., 2004, Pesquita et al., 2009, Azuaje et al., 2005, Li et al., 2010, Pesquita et al., 2007, Ravasi et al., 2010, Washington et al., 2009, Li et al., 2013, Teng et al., 2013, Yang et al., 2012]. Since different biomedical researchers may annotate the same or similar gene function with different but semantically similar GO terms based on their research findings, an accurate measure of semantic similarity of GO terms is critical for accurate measurement of gene functional similarities.

Gene Ontology is represented as a directed acyclic graph (DAG), in which nodes correspond to terms and edges represent relationships between the terms. It defines several relationships between nodes, with the important ones being: is_a, part_of, regulates, negatively_regulates, and positively_regulates [Consortium, 2000]. In the graph of Gene Ontology, there is a node specified as the root. For every node in the ontology, there exists at least on e path pointing from the root to the node. Every node in the path is called an ancestor of the node, and the ancestor that immediately precedes the node is called he parent of the node. Conversely, if a node is a parent of another node, the other node is called a child of the parent. There could be multiple paths from the root to the node. Consequently, a node may have multiple parent nodes and vice versa. Figure 3.1 gives an snippet extracted from Gene Ontology. For

instance, we can see from the figure that terms in Gene Ontology can have multiple parents. The term "regulation of cellular process" (GO:00500794) in Figure 3.1 has two parents, "regulation of biological process" (GO:0050789) and "cellular process" (GO:0009987) respectively.

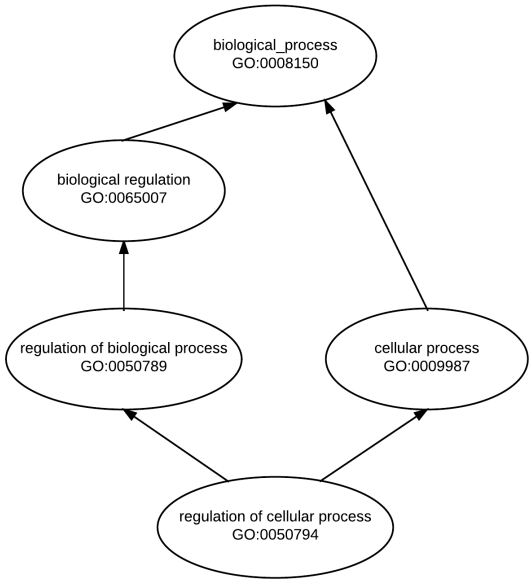

Figure 3.1: An snippet extracted from Gene Ontology

Gene Ontology (GO) [Consortium, 2000] aims at fulfilling the requirements of research community for correct descriptions of gene products, e.g. proteins. The role that a protein performs in a biological process, molecular function or cellular component is defined by assigning a GO term to the protein. This annotation of proteins with GO terms makes a valuable knowledge base that integrates the knowledge of many scientists on a single platform which can be processed by computers using modern semantic computing methods. There is an increasing gap between the availability of knowledge and the methods and tools to process and use it. The rapid growth of these resources is both a challenge and opportunity for biology and com-

puter science professionals. A key problem is to computer similarity between ontology terms and between gene products.

While those existing studies have proposed different methods to measure the semantic similarity of GO terms, they all have their limitations. In general, there are three types of methods for measuring the semantic similarity of GO terms: node-based [Schlicker et al., 2006, Resnik, 1999, Lin, 1998, Jiang and Conrath, 1997], edge-based [Cheng et al., 2004, Pekar and Staab, 2002, Wu et al., 2005, Li et al., 2013], and hybrid [Wang et al., 2007, Pesquita et al., 2009, Teng et al., 2013, Yang et al., 2012] methods. The three most cited representative methods [Resnik, 1999, Lin, 1998, Jiang and Conrath, 1997] were originally designed to measure the semantic similarity of natural language terms. While they have been widely adopted by bioinformatics researchers to measure the semantic similarity of GO terms, each of them has its own limitations. In 2007, Wang [Wang et al., 2007] proposed a new measure of the semantic similarity of GO terms: this new hybrid method considers both the GO structure and the semantic content (biological meaning) of the GO terms in measuring the semantic similarity of GO terms, and many studies [Xu et al., 2008, Pesquita et al., 2009, Ravasi et al., 2010, Washington et al., 2009] have shown the superiority of this hybrid method. It has been widely accepted by biomedical researchers [Pesquita et al., 2009] since it was published.

3.1 Semantics of GO Terms

For decades, people have been searching for answers in how life works. What a biological process, a gene or protein really does, where, when and how it is related to other parts of the life still remain a big challenge for human beings. There are tons of gene products whose functions we have no idea of without doing the actual

heavy experiments. An important question is then raised up: how to know if a new gene product is discovered? To be able to answer this question, we need to know the relation between gene products based on which we can distinguish one from another. To be able to figure out the relation, we need to figure out the what exactly is a gene product. Another question comes up: how to represent a gene product? Techniques including ontology, DNA sequencing have been extensively exploited to aid biomedical researchers answering the above questions. DNA sequencing can be used to determine the precise order of nucleates within a DNA molecule. This technique has become indispensable for biological research. Even gene sequencing provides an precise representation for gene products, it suffers from its high cost and the difficulties to analyze gene sequences. Fortunately, ontology provides a succinct yet powerful representation of gene products that enables us to study the relation between gene products based on which we can distinguish one from another. In ontology, the specification of conceptualization divides gene products into biological entities. From experiments, researchers can figure out what processes the gene product is involved in and which parts it belongs to. Therefore, a gene product can be annotated with these separate biological entities or, in other words, these associated biological entities represent the gene product. With this compact representation, we can discover relations between gene products by studying their corresponding representations.

The GO Consortium constructed a Directed Acyclic Graph (DAG) of GO terms representing biological entities to capture the relation between them. These manually selected terms (biological entities) have greatly expedited the process of gene product analysis and already been applied in numerous fields such as medical diagnosis, biotechnology, bioinformatics, etc.

3.2 Limitations of Current Methods

Generally speaking, there are three types of measurement on semantic similarity measurement between terms in the ontology: node-based, edge-based, and hybrid.

Node-based measures (e.g. Resnik's [Resnik, 1999], Lin's [Lin, 1998], Jiang and Conrath's [Jiang and Conrath, 1997], Schlicker's [Schlicker et al., 2006]) rely mainly on ***Information Content (IC)*** of the terms to represent their semantic values; IC of a term is derived from the frequency of its presence (including the presence of its children terms) in a certain corpus (e.g. SGD database, Gene Ontology database). Resnik's [Resnik, 1999] method concentrates only on the MICA of the compared terms, but ignores the locations of these terms in the graph, e.g., a term's distance from the root of the ontology, and the semantic impact of other ancestor terms. A term's distance to the root of the ontology shows the specialization level of this term in human perception. If a term is far from the root in the ontology, it means researchers know more details about this term and the meaning of the term is more specific. On the other hand, if a term is closer to the root of the ontology, it means the term is a more general term, such as cellular process or metabolic process, which does not provide too much details about the related entities. Ignoring the specialization level of a term in the ontology is the principal reason that the semantic similarity obtained by these methods is inconsistent with human perception; they suffer from "shallow annotation" problem [Sevilla et al., 2005, Li et al., 2010, Wang et al., 2007] in which the semantic similarity of terms near the root of the ontology are sometimes measured very high.

Edge-based approaches [Cheng et al., 2004, Pekar and Staab, 2002, Wu et al., 2005, Li et al., 2013] are based on the length of graph paths connecting the terms

being compared. Some edge-based approaches [Pekar and Staab, 2002] treat all edges equally, ignoring the levels of edges in the ontology. This simple equal-edge-based approach also suffers from "shallow annotation" because based on this approach, the semantic similarity of two terms with a certain graph distance near the root would be equal to the semantic similarity of two terms with the same graph distance but away from the root. To address the "shallow annotation" problem, other edge-based methods [Cheng et al., 2004, Wu et al., 2005, Li et al., 2013] assign different weights to the edges at the different levels of the ontology, assuming that the edges at the same level of the ontology have the same weight. However, the terms at the same level of the graph do not always have the same specificity because different gene properties demand different levels of detailed studies. It means the edges at the same level of the graph but in different branches do not necessarily have the same weights.

The hybrid method [Wang et al., 2007] considers both the structure and the semantics of terms at different ontological levels. However, this method uses two semantic contribution factors, obtained from empirical study of gene classification of certain species, to calculate the semantic values of terms. Semantic contribution factors obtained by empirical studies on genes from certain species may not be optimal for measuring the functional similarity of genes in other species. A recent study [Yang et al., 2012] has proposed to consider the interaction between the descendants of the terms in computing semantic similarity between them. However, this is predominantly an add-on to other existing methods and thus still inherits their drawbacks.

We propose a new measurement of semantic similarity of terms in biology domain that can efficiently utilize ontology knowledge as well as corpus statistics. We used Gene Ontology as the knowledge resource upon which we validated the proposed approach.

3.3 Aggregate Information Content (AIC) Based Method

We propose a novel method — *Aggregate Information Content* (AIC) based similarity measurement to measure the semantic similarity of terms. The proposed method considers the aggregate contribution of the ancestors of a term (including this term) to the semantics of this term, and takes into account how human beings use the terms to annotate genes. We use a term's IC value, as defined before (Equations 2.1, 2.2, 2.3), to represent their semantic contribution values. Given the fact that terms at upper levels (more general terms) of ontology graph are less specific than those at lower levels, we define the ***knowledge*** of a term t as,

$$K(t) = 1/IC(t) \tag{3.1}$$

Unlike the weight in Method D, which represents the semantic contributions of ancestor terms using contribution factors obtained from empirical study, this $K(t)$ incorporates the statistical distribution of terms in the entire Ontology. The higher a term dwells in the ontology, the more we know about this term (with more knowledge, i.e., K(t)). Thus, we would say this newly defined $K(t)$ represents how much people have studied term t, thus the ***knowledge*** of term t. We further propose a logarithmic model to normalize $K(t)$ into a ***semantic weight*** $SW(t)$:

$$SW(t) = \frac{1}{1 + e^{-K(t)}} \tag{3.2}$$

We then compute ***semantic value*** $SV(a)$ of the term a by adding the semantic weights of all its ancestors (i.e., aggregating semantic contribution of the ancestors).

$$SV(a) = \sum_{t \in T_a} SW(t) \tag{3.3}$$

where T_a is the set of all of its ancestors including a itself. We define the semantic similarity between terms a and b, based on their aggregate information content (AIC), as follows.

$$sim(a,b) = \frac{\sum_{t \in T_a \cap T_b} 2 * SW(t)}{SV(a) + SV(b)} \tag{3.4}$$

where $SW(t)$ is the semantic weight of term t defined in Equation 3.2, and SV(t) is the semantic value of term t defined in Equation 3.3. Aggregating the semantic contribution of all ancestor terms implicitly factors in the position of the term in the Gene Ontology graph, and overcomes the weakness of the MICA based approaches.

This proposed AIC method is based on two major observations: (1) In general, the dissimilarity of terms near the root (more general terms) of the ontology graph should be larger than that of the terms at a lower level (more specific terms); (2) the semantic meaning of one term should be the aggregation of all semantic values of its ancestor terms (including the term itself). The first observation follows the human perception of term semantic similarity at different specialization levels of the ontology. The second observation agrees with how human beings use the term to annotate genes.

We demonstrate how to use the AIC method to compute the semantic similarity between two terms, GO:0050794 and GO:0050789, shown in Figure 3.2. (All GO DAGs are obtained from the Web tools in the popular G-SESAME Website [Du et al.,

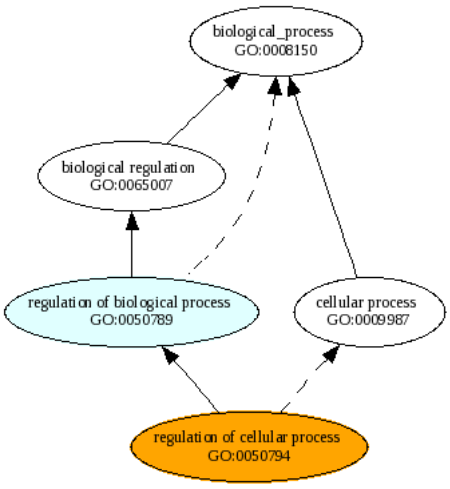

Figure 3.2: GO Graph containing terms GO:0050794 and GO:0050789

Go Terms	IC value	SW value
0050794	1.2461	0.6905
0050789	0.9906	0.7329
0065007	0.9403	0.7434
0009987	0.2610	0.9788
0008150	0	1

Table 3.1: IC values & Semantic Weights of GO terms

2009].) First, we obtain the IC values of all related GO terms from the GO database. The results are shown in Table 3.3. We note that many studies [Ovaska et al., 2008, Li and Lu, 2008, Smialowski et al., 2010] used GOSim R package [Froehlich et al., 2007] to obtain the IC information for all related GO terms. However, IC values in GOSim R package are not always derived from the latest GO database. Due to continuous evolution of the GO database, IC values and semantic similarities of GO terms may change over time with the change of the GO database content. In addition, GOSim R package is hard to be integrated with the popular G-SESAME Website. Another widely used R package called GOSemSim [Yu et al., 2010] also suffers those issues. Therefore, we chose to calculate the IC values of GO terms directly from the GO database release. Second, we calculate the semantic weight for each GO term using Equation 3.2. Finally, we use Equation 3.3 and Equation 3.4 to get the semantic similarity of GO terms GO:0050794 and GO:0050789 as $sim_{GO}(0050794, 0050789) = 0.748$.

3.4 Validation of AIC

3.4.1 Comparison Analysis Based on Correlation with Human Perception

From human perspective, we know that two GO terms at higher levels of the gene ontology should have larger dissimilarity than two GO terms with the same graph distance at lower levels. Our AIC method is compatible with this observation in that two GO terms with the same graph distance at the lower levels of the gene ontology usually share more common ancestors. Therefore, the semantic similarity of GO terms obtained by our AIC method is consistent with human perception as shown

in an illustrative example from our experimental results in Figure 3.3 and Table 3.2.

Consider the two GO terms GO:0005739 and GO:0005777 as shown in Figure 3.3. The semantic similarity values SW, obtained by Methods A, B, C, D and AIC are shown in Table 3.2. These two very specific GO terms have only one different ancestor term GO:0042579; the semantic similarity between them should be very high. However, the semantic similarity values obtained by Method B [Lin, 1998] and Method C [Jiang and Conrath, 1997] fail to exhibit this expected behavior while Method D [Wang et al., 2007] and the proposed AIC method correctly exhibit this expected behavior. This observation reinforces our previous contention that use of MICA alone in computing similarity is not sufficient because of loss of important information. The semantic similarity values obtained by Method A are not normalized; hence, it is hard to determine the relative similarity levels without reviewing all pair-wise semantic similarity values in the GO database. Accordingly, we have exluded Method A from this comparison study.

Dataset	Method	**Similarity**
SW(GO:0005739, GO:0005777)	A	1.047
	B	0.424
	C	0.260
	D	0.797
	AIC	0.902
SW(GO:0044424, GO:0005622)	A	0.430
	B	0.918
	C	0.928
	D	0.845
	AIC	0.898
SW(GO:0044444, GO:0005737)	A	0.821
	B	0.879
	C	0.815
	D	0.879
	AIC	0.939

Table 3.2: Semantic similarity values of GO term pairs obtained by different methods

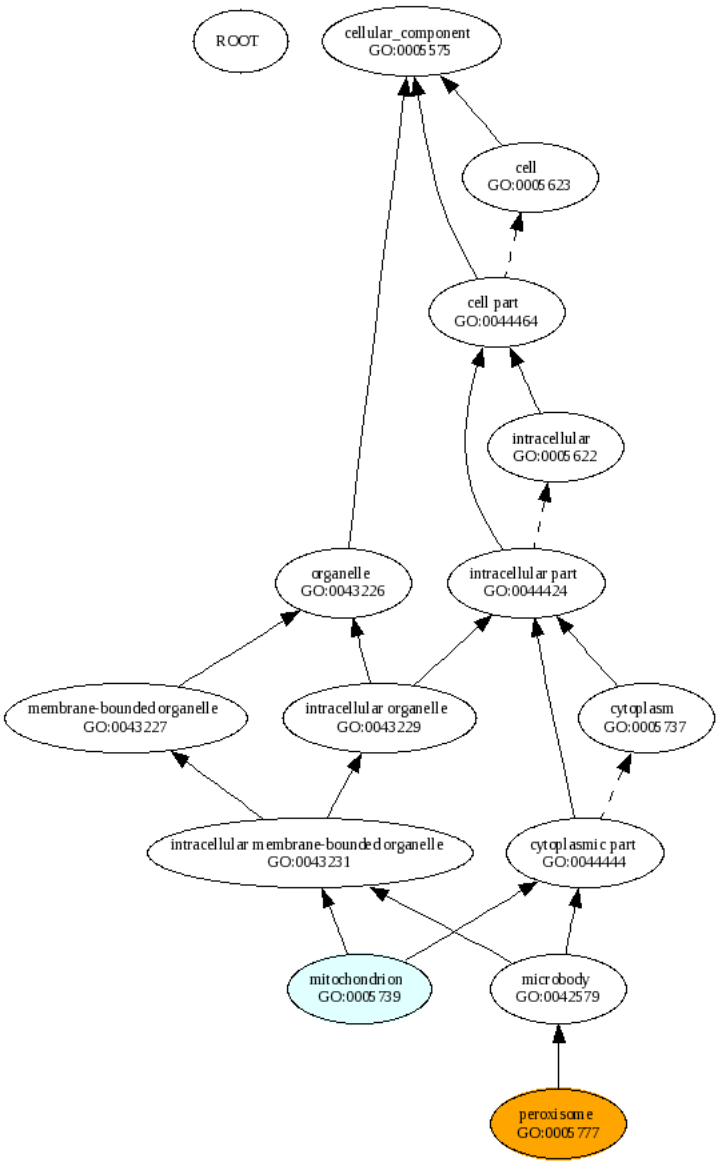

Figure 3.3: GO graph of terms GO:0005739 and GO:0005777

Now, we check whether all these semantic similarity measurement methods agree with the human perspective aforementioned: two GO terms at higher levels of the gene ontology should have larger dissimilarity than two GO terms with the same graph distance at lower levels. We calculate the semantic similarity between GO:0044424 and GO:0005622 (Group 1) and the semantic similarity between GO:0044444 and GO:0005737 (Group 2). The semantic similarity values are shown in Table 3.2. These two groups of GO terms have similar structure in the GO graph except group 1 is closer to the root of the GO graph. Based on human perception, the semantic similarity of GO terms in group 1 should be less than that in group 2 since GO terms in group 2 are at a lower level of the GO graph. However, only methods A, D and our AIC method satisfy this property. The semantic similarity values obtained by methods B and C are inconsistent with the human perception because these two methods do not consider the specialization level of two terms' LCA in the semantic similarity measure. The "shallow annotation" problem is clearly shown in these experiments.

3.4.2 Comparison Analysis Based on Correlation with Gene Expression Data

[Wang et al., 2004, Xu et al., 2008, Sevilla et al., 2005] demonstrates that there is a high correlation between gene expression data and the gene functional similarity obtained from GO term similarities, i.e., genes with similar expression patterns should have high similarity in GO based measures because they should be annotated with semantically similar GO terms. We use the correlation of genes obtained from gene expression data to validate the gene functional similarities obtained by GO based similarity measures. As in many existing studies [Li et al., 2010, Heyer et al.,

1999, Jiang et al., 2004, Gibbons and Roth, 2002], we use gene expression data from Spellman dataset [Spellman et al., 1998], which comprises of 6178 genes, to obtain the gene correlation patterns. In the next two subsections, we provide comparison of our method (AIC) with the state-of-the-art current methods: Method A [Resnik, 1999], Method B [Lin, 1998], Method C [Jiang and Conrath, 1997], and Method D [Wang et al., 2007] in terms of GO term semantic similarity and gene functional similarity.Most newly proposed methods are different variants of methods A, B, and C which are widely used as benchmark methods for measuring the semantic similarity of GO terms.

The functional similarity between gene products can be defined as the maximum or average semantic similarity values over the GO terms annotating the genes respectively. We use the AVE method as follows,

$$sim_{AVE}(g_1, g_2) = \underset{\substack{t_1 \in anno(g_1) \\ t_2 \in anno(g_2)}}{average}\ sim(t_1, t_2) \tag{3.5}$$

where $annotation(g)$ is the set of GO terms that annotates gene g. Although some studies [Wang et al., 2007, Sevilla et al., 2005] use the MAX method to compute the functional similarity of genes, people [Xu et al., 2008] found that the AVE method is more stable and less sensitive to outliers. In addition, the AVE method is more compatible with our original objective of capturing all available information while the MAX method takes the most significant GO terms ignoring the contributions of others. Hence, we use the AVE method in the experiment.

We first use Pearson's correlation to compute the gene expression similarity with the Spellman dataset [Spellman et al., 1998]. Then, we calculate the correlation between the functional similarity of these genes obtained from BP ontology and the gene expression similarity. The objective is, as stated in [Wang et al., 2004], to

test the hypothesis that pairs of genes exhibiting similar expression levels which are measured by the absolute correlation values in gene expression data tend to have high functional similarities between each other. The average of correlation coefficients between genes within an expression similarity interval estimates the mean of the statistical distribution of correlations; and it shows the underlying trend that connects gene expression similarity with functional similarity. We split the gene pairs into groups with equal intervals according to the absolute gene expression correlation values between gene pairs, as in previous studies [Li et al., 2010, Xu et al., 2008, Wang et al., 2004, Sevilla et al., 2005], and then compute Pearson's correlation coefficient between the mean of gene functional similarities and the mean of gene expression correlation values in each group. We split gene pairs into 4-13 groups respectively to avoid under-fitting and over-fitting problems [Courrieu et al., 2010]. We again compare the results obtained using four existing methods (Methods A, B, C and D) and those obtained using our AIC method, as shown in Table 3.3. The experimental results show that our AIC method generally outperforms other four methods with the highest correlation coefficients between gene functional similarity and gene expression similarity in most cases. Method D and C also showed excellent correlation between GO based functional similarity and the gene expression similarity. Method A and B did not perform well in this experimental study, with Method A being the worst.

3.4.3 Computational Efficiency of the AIC Method

While methods D and AIC show superiority to other three methods in agreement with human perception and in correlation with gene expression data, Method D requires computation of the S-value of a node in a DAG by doing a breadth first search starting from the node and exhausting the subtree of the node in the DAG

Groups	Method A	Method B	Method C	Method D	**Proposed AIC**
4	0.614	0.789	0.930	0.929	**0.966**
5	0.561	0.717	**0.889**	0.802	0.850
6	0.413	0.569	0.700	0.745	**0.774**
7	0.519	0.622	**0.761**	0.725	0.733
8	0.496	0.597	0.675	0.706	**0.714**
9	0.417	0.659	0.664	0.745	**0.778**
10	0.403	0.620	0.730	0.733	**0.772**
11	0.419	0.665	0.691	0.725	**0.761**
12	0.246	0.485	0.722	0.716	**0.782**
13	0.321	0.525	0.715	0.709	**0.791**

Table 3.3: Pearson's correlation coefficients between gene expression data and gene functional similarities obtained by different semantic similarity measurement methods

(S-values are not stored at the node); this is a major computation cost of method D. In the proposed AIC method, the similarity values are precomputed using the DAG (the GO graph and its relationships are static and stored as the aggregate information content of the node) and are stored at the node; Therefore, this method significantly reduces computational cost without the need of traversing the entire graph. which is the primary reason for the computational efficiency of AIC method over Metod D. We use the execution time of computing the functional similarities of a large number of gene pairs to evaluate the computation efficiency of our proposed AIC method. In this experiment, we use methods D and AIC to compute the functional similarities of three sets of gene pairs. The experiment was conducted on a Linux box with a i7-2600K CPU @ 3.40GHz, 32G memory. The execution time are shown in Table 3.4 as an average of the time in ten tests on each number of gene pairs. As demonstrated by the experimental results, method AIC is considerably faster than method D.

3.4.4 Advantages of AIC Method

Experimental results in the above section demonstrate the superiority of the proposed AIC method over the representative ones, Method A [Resnik, 1999], Method B [Lin, 1998], Method C [Jiang and Conrath, 1997] and Method D [Wang et al., 2007]. Method AIC is characterized with the following unique features:

- AIC shows advantages over Method A by taking into account the structural difference.

- AIC does not suffer from "shallow annotation" as in Method B and Method C. Note that, in Equation 3.4 the denominator is smaller when terms are annotated at the top levels, i.e., the equal difference on the numerator will result in a larger difference in the semantic similarity value. Thus, the semantic similarity value of two terms at top levels is less than that of two terms with the same graph distance at lower levels. This is consistent with human perspectives.

- AIC exhibits high correlation coefficient between the gene expression similarity and the GO based functional similarity.

- AIC is computationally significantly faster than the popular hybrid Method D since the information content values can be precomputed. It does not use the empirically determined semantic contribution factors in semantic similarity computation.

In summary, the proposed AIC method is very promising in that it outperforms all existing state-of-the-art methods in terms of consistency with human perception, correlation with gene expression data and computational efficiency.

	Execution Time (seconds)		
# of Gene Pairs	200	500	2000
Method D	173	3506	36123
Method AIC	56	261	7632

Table 3.4: Computation Efficiency of Methods D and AIC

3.5 Integration with G-SESAME Web Services

Due to the high demand for computing GO semantic similarity and gene functional similarity by biomedical researchers, a number of web services, such as ProteInOn (`http://lasige.di.fc.ul.pt/webtools/proteinon`) [Faria et al., 2007], FunSimMat (`http://www.funsimmat.de`) [Schlicker and Albrecht, 2008], GOToolBox (`http://genome.crg.ex/GOToolBox`) [Martin et al., 2004], and G-SESAME (`http://bioinformatics.clemson.edu/G-SESAME`) [Wang et al., 2007] have been developed. All of them are very convenient and easy to use. However, none of these tools, except G-SESAME, provides the visualization, batch-mode support, and Web-based APIs simultaneously. These enhanced services are very important and useful for biomedical researchers to conveniently and efficiently run their applications. G-SESAME is a set of on-line tools to measure the semantic similarities of Gene Ontology (GO) terms and the functional similarities of gene products, and to discover biomedical knowledge through GO database. The set of tools have been used more than 72.3 million times by researchers from 252 organizations since 2006. We augmented and extended the original G-SESAME website by incorporating our proposed AIC method, and also implemented the other three popular methods, A, B and C, to allow users to select the appropriate method at their own interests. The redesigned G-SESAME Website has the following characteristic features:

1. It provides a list of user-friendly, easy-to-use Web services for researchers to use.

2. It provides several state-of-the-art semantic measurement methods at the same place. Users can select different methods to measure the semantic similarity of GO terms and functional similarity of genes, and compare their measurement results.

3. It provides Web-based visualization to allow users to inspect the locations of the GO terms within the GO graph and visually determine their semantic similarity.

4. It provides batch mode support to allow users to measure the semantic similarities of a group of GO terms or functional similarities of a group of genes.

5. It provides a list of Web-based APIs to allow users to easily integrate these Web services into their own applications.

We here show the some of the key functionalities of the redesigned G-SESAME. Figure 3.4 shows the redesigned interface of the original G-SESAME method [Wang et al., 2007]. Users can simply input the two GO term to be compared and the weights of the "is-a" and "part-of" relationship. The result page shown in Figure 3.5, returns the semantic similarity (highlighted in red) between the two GO terms the user specified, the definition of each GO term and their structural information in the Gene Ontology graph. Users can also choose to use the proposed AIC method and other statistical methods [Resnik, 1999, Lin, 1998, Jiang and Conrath, 1997] as shown in Figure 3.6, with the result page being Figure 3.7. G-SESAME also provide the functionality of calculating two sets of GO terms. As shown in Figure 3.8 and Figure 3.10, users can choose to upload two sets of GO terms in the format of text files. The corresponding result pages are shown in Figure 3.9 and Figure 3.11, respectively.

G-SESAME also provides the function to compute the functional similarity between two genes. Figure 3.12 demonstrates the provided user interface for users to input the two gene names to be compared, the corresponding ontologies, species, data sources and the evidence codes. The result page is shown in Figure 3.13.

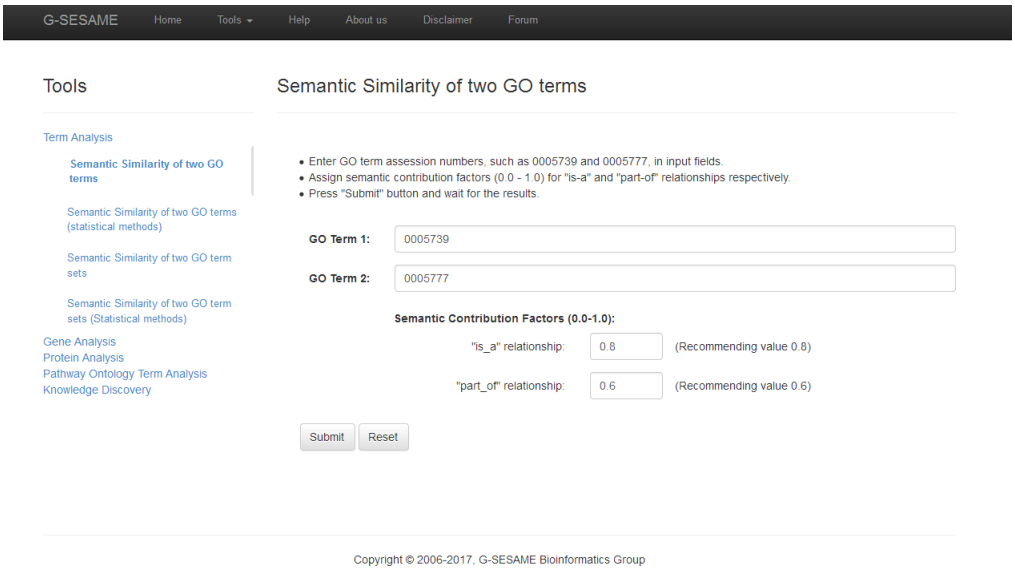

Figure 3.4: Semantic similarity between two GO terms using [Wang et al., 2007]'s method

The proposed AIC method to measure the semantic similarity of terms on Gene Ontology was reported in [Song et al., 2014, Song et al., 2013]. While ontologies such as the Gene Ontology were specially designed for a specific domain, we need a more general ontology that covers terms in our daily life if we want to analyze text documents which could cover terms in multiple domains. WordNet is the most widely used ontology that contains regular words to perform NLP tasks including document semantic analysis. In the next section, we introduce an ontology-based approach to derive a hybrid word representation that combines context information and pre-established knowledge in WordNet to derive the text similarity measurements.

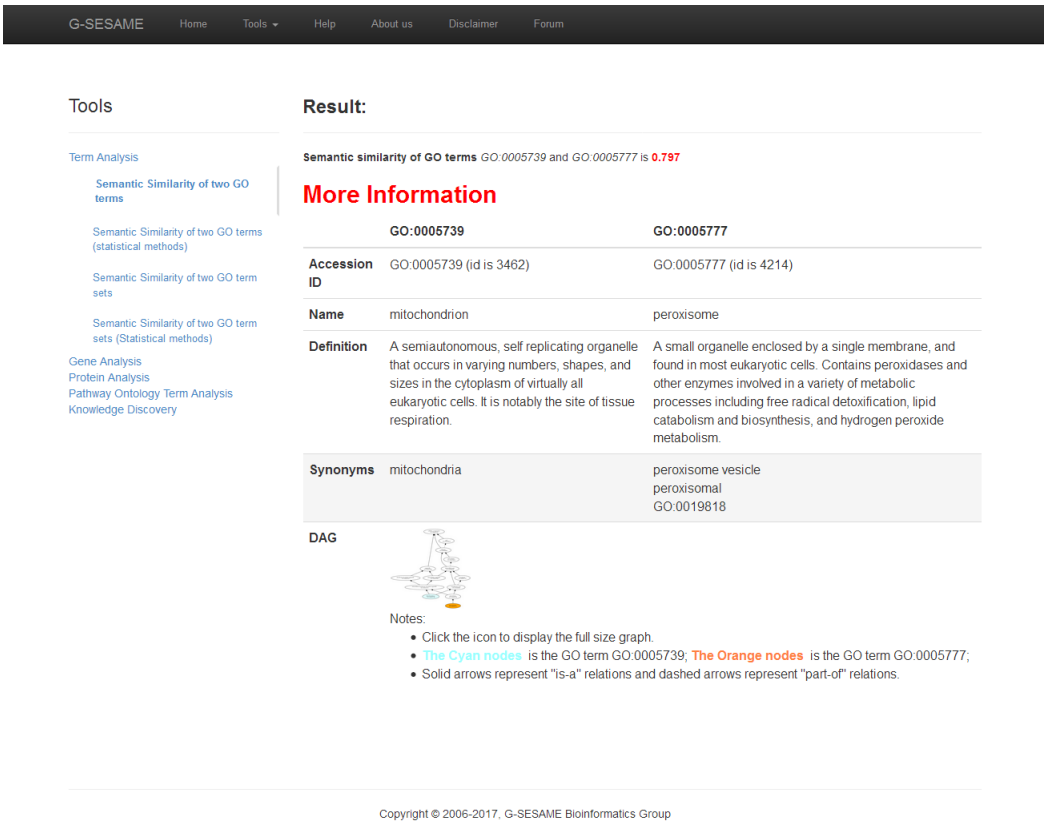

Figure 3.5: Result page of semantic similarity between two GO terms using [Wang et al., 2007]'s method

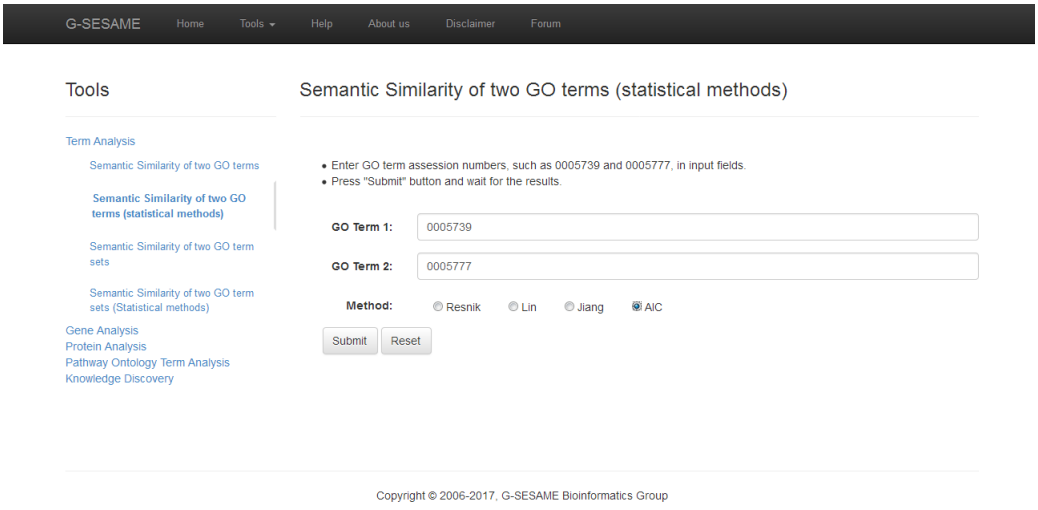

Figure 3.6: Semantic similarity between two GO terms using statistical methods

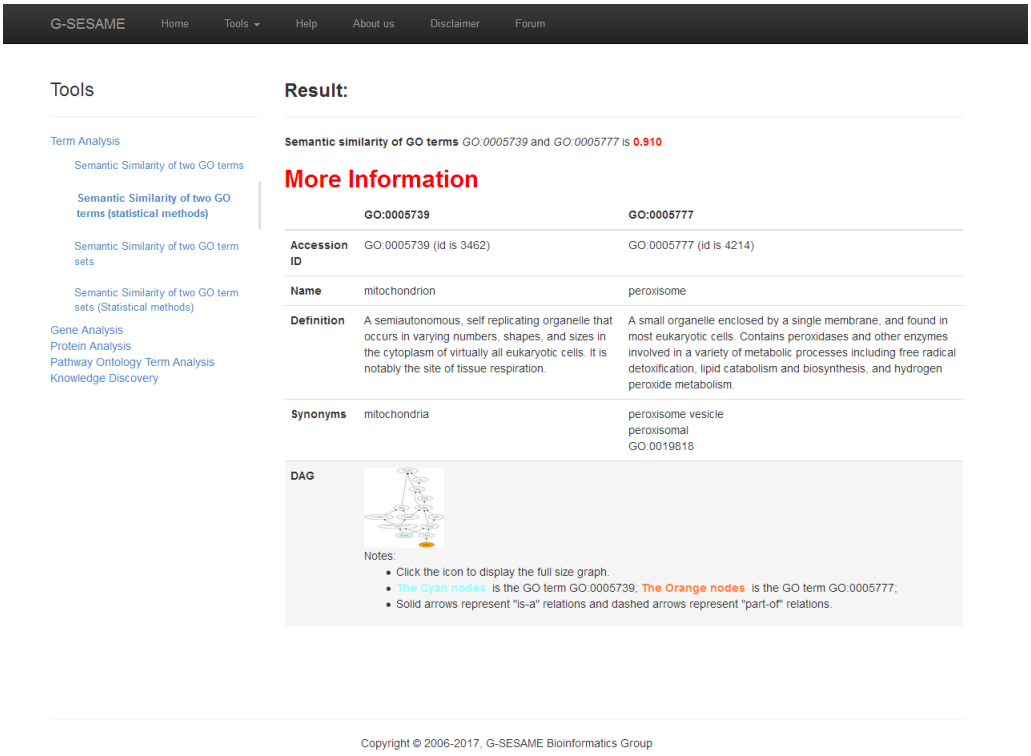

Figure 3.7: Result page of semantic similarity between two GO terms using statistical methods

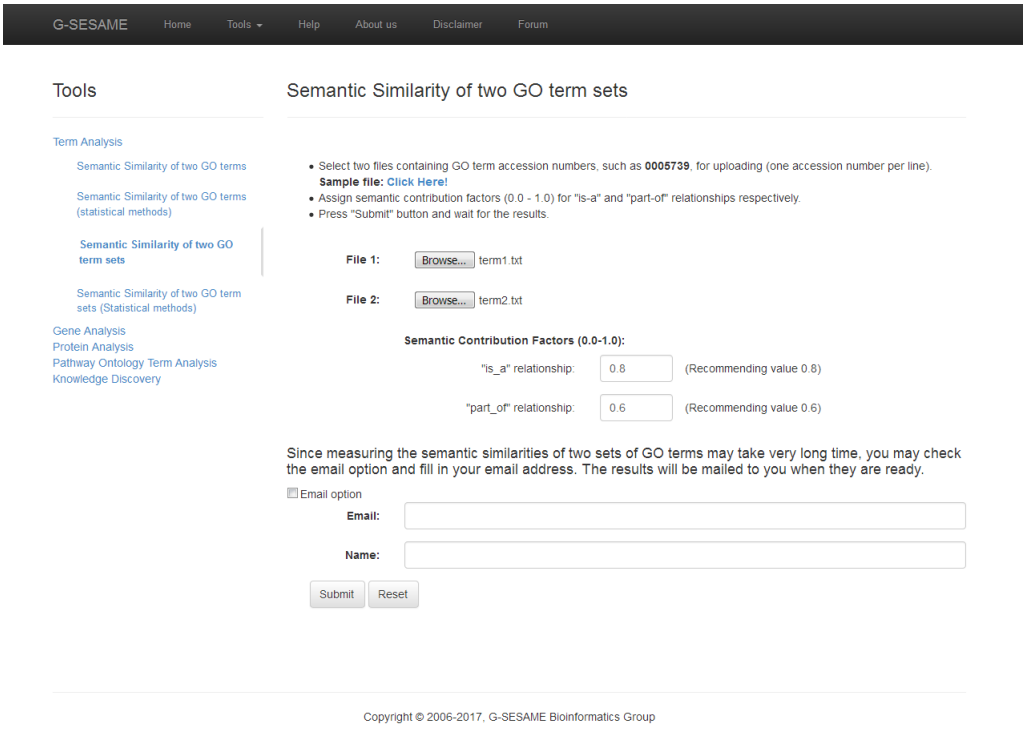

Figure 3.8: Semantic similarity between two GO terms using [Wang et al., 2007]'s method

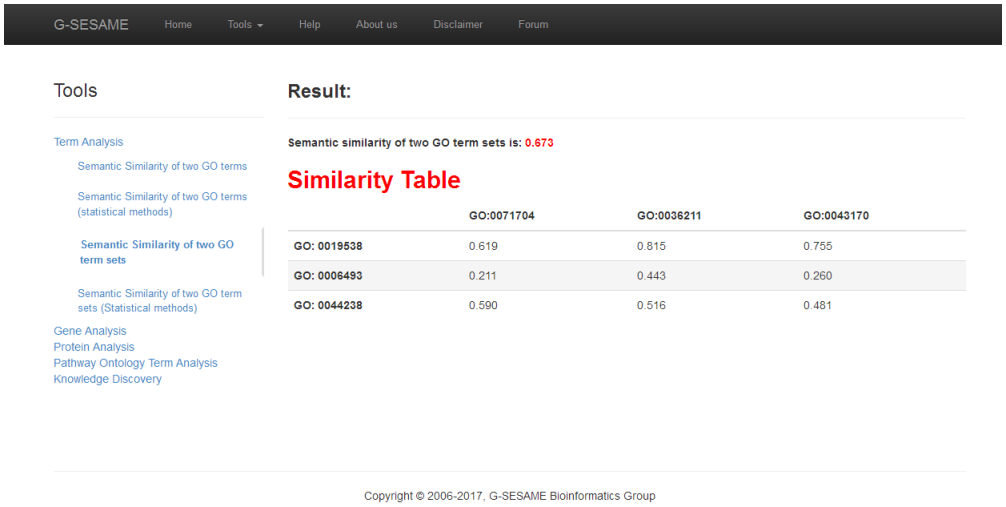

Figure 3.9: Result page of semantic similarity between two GO terms using [Wang et al., 2007]'s method

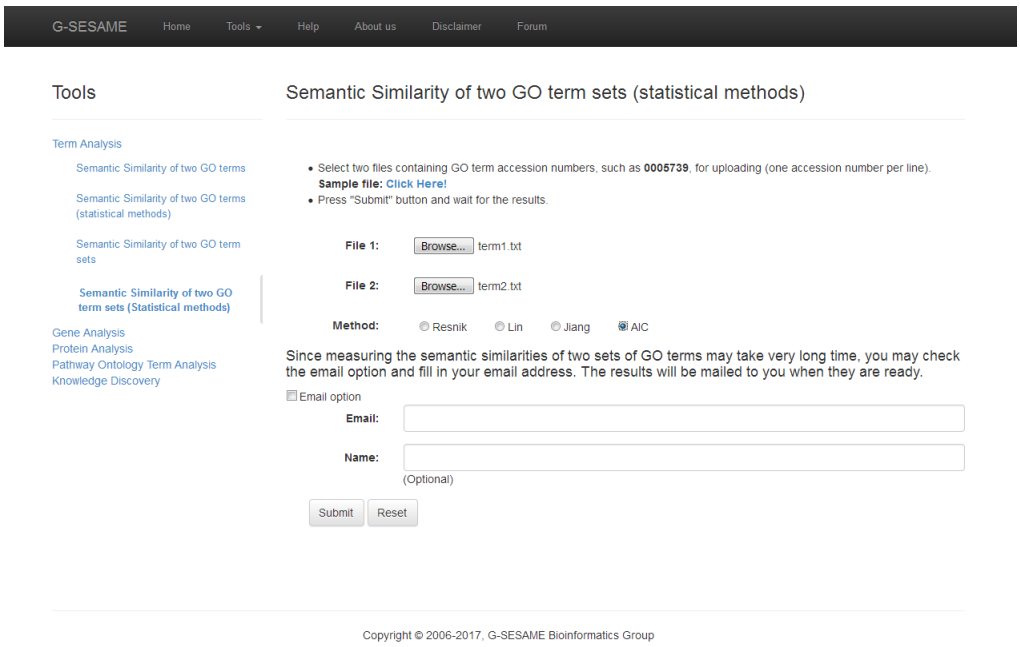

Figure 3.10: Semantic similarity between two sets of GO terms using statistical methods

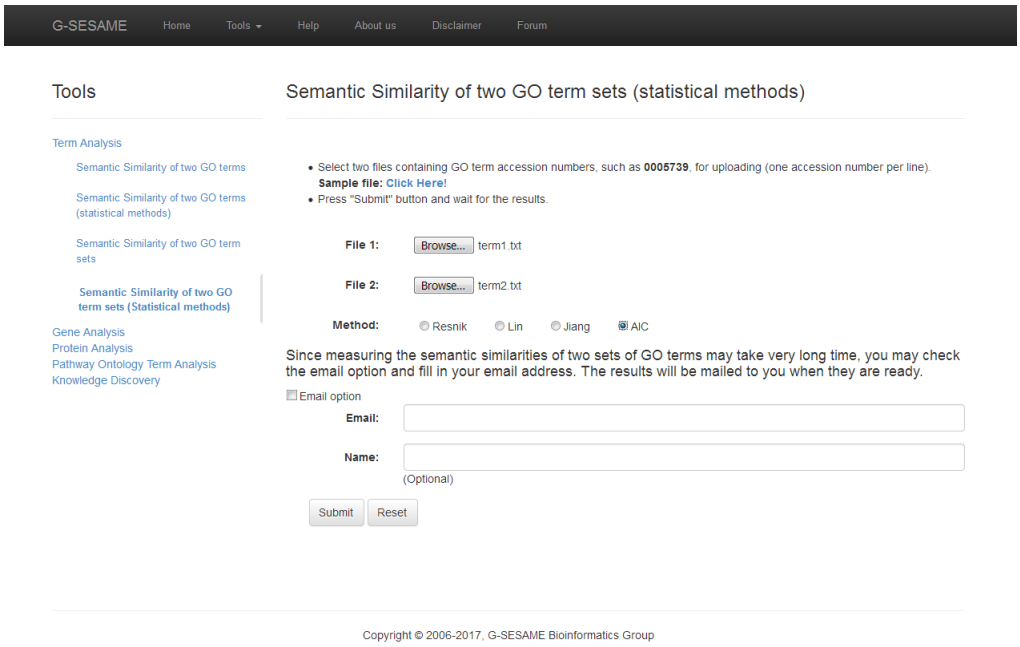

Figure 3.11: Result page of semantic similarity between two sets of GO terms using statistical methods

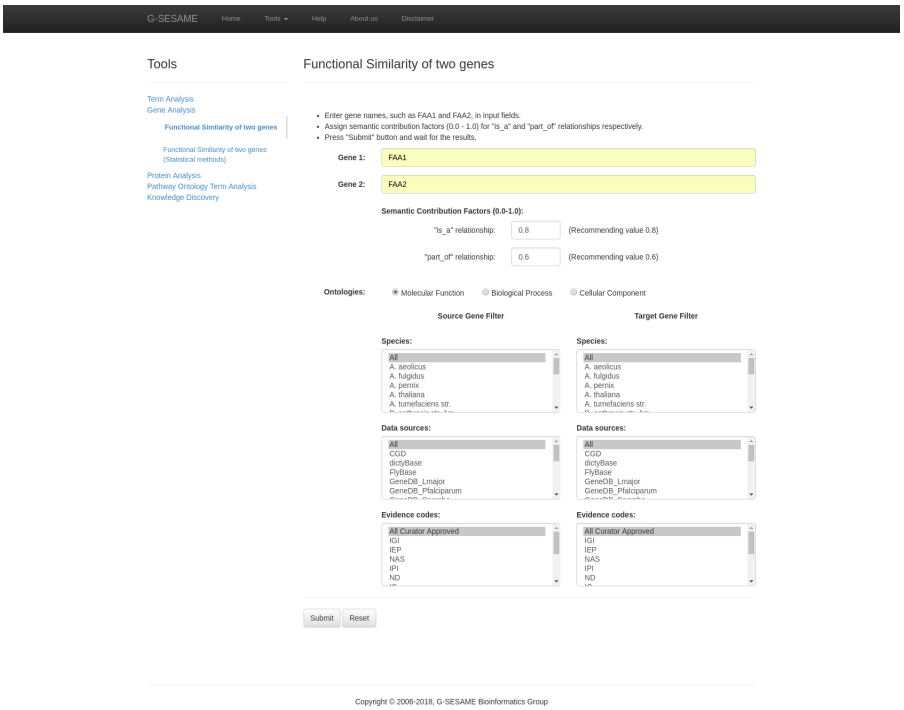

Figure 3.12: Functional similarity between two genes

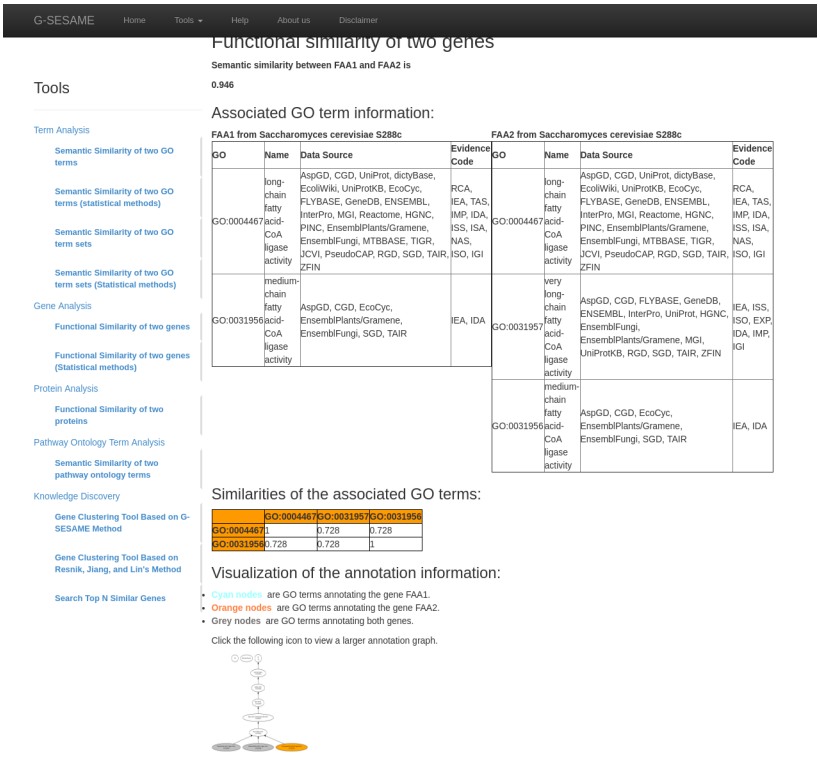

Figure 3.13: Result page of functional similarity between two genes

Chapter 4

Semantic Similarity Analysis of Words/Text Documents

4.1 Text Similarity

Text similarity measurements are becoming increasingly important in tasks such as information retrieval, text classification, document clustering, topic detection, topic tracking, questions generation, question answering, machine translation, text summarization, etc [Seco et al., 2004, Bollegala et al., 2007, Corley and Mihalcea, 2005, Varelas et al., 2005, Salton and Buckley, 1988, Rocchio, 1971, Lesk, 1986, Leacock and Chodorow, 1998, Leacock et al., 1998, Budanitsky and Hirst, 2001, Salton et al., 1997, Voorhees, 1993, Xu and Croft, 1996, Varelas et al., 2005, Corley and Mihalcea, 2005, Seco et al., 2004]. The vector model in [Salton and Lesk, 1968] is perhaps one of the earliest applications of text similarity, where the document most relevant to the input query is determined by ranking documents in a collection in descending order of their similarity to the given query. The typical approach to computing the similarity between two text segments is to adopt a lexical matching

method, producing a similarity score based on the number of words that occur in both input segments. Improvements including stemming, stop-word removal, longest subsequence matching, part-of-speech (POS) tagging, as well as various weighting and normalization schemes [Salton and Buckley, 1988]. However, these lexical matching similarity methods fail to identify the *semantic similarity* of texts. For example, they fail to identify any kind of connections between the two text segments, "I have a dog" and "I have a pet", while there is an obvious similarity between them. While Latent Semantic Analysis (LSA) [LANDAUER and DUMAIS, 1997, Landauer et al., 1998] is used as semantic similarity measurements [Voorhees, 1993, Xu and Croft, 1996] to find similar terms in large text collections by including additional related words, the "black-box" effect of LSA does not follow any deep insights into why some terms are chosen as similar during the singular value decomposition process.

Given two input text segments, we want to automatically drive a score that can indicate the similarity between the two text segments at a *semantic* level, going beyond the traditional lexical matching methods. As text segments consists of words, we consider this problem as finding word representations that can capture not only the relations between words, but also the local context information, which contains the roles played by the various entities involved in the interactions described by the two text segments. Traditional machine learning systems require large corpus of data to train the text model. The performance of the trained models usually highly rely on the quality of the training corpus. The labor and resources to obtain high quality could be very expensive, let alone the rapid generation of the large amounts of documents. Moreover, in terms of short texts, such as microblogs and tweets, the information contained in the text might be relatively limited, it is hard to determine word semantics by the context alone. Therefore, it is necessary to incorporate knowledge from other sources, such as ontologies to analyze text. In this

section, we introduce methodologies to measure the semantic similarity among text documents incorporating domain specific knowledge. Specifically, we use WordNet as the ontology, from which we retrieve existing knowledge to capture more semantics. An ontology-based neural network model is proposed to measurement the semantic similarity of text documents. This model first generates hybrid word representations that not only can extract knowledge from ontologies such as WordNet, but also can utilize the context information within the text. Document representations are then formed from the generated hybrid word representation to calculate semantic similarities between document.

4.2 Enriching Word Representations with Ontology

Domain knowledge consists of information about the data that is already available either through discovery process or from a domain expert. In text mining, it is very important to capture domain knowledge as it can make texture patterns more visible and constrain the search space as well as the rule space. In a classification or predictive modeling task, domain knowledge could help improve learning efficiency as well as the quality of the learned model. It could also be used to initialize a knowledge structure and make the discovered knowledge more interpretable. Traditional document representation techniques focus on using word co-occurrence models, such as Vector Space Model (VSM), with an assumption that related words will often appear in the same documents. For instance, “fumble”, “touchdown”, “score”, “intercept”, etc. are likely to appear together, while “fishing”, “papers”, “speeding”, and “city” would probably not appear together.

VSM represents a text document by a vector of frequencies of terms appearing in the document. The similarity between two text documents is determined by the cosine coefficient between the two document vectors. One of the major drawbacks of VSM approach is its inability to handle the polysemy and synonymy phenomena of natural language. In terms of polysemy, the same word could be used for different concepts, e.g. "Tigers" in National Geographic probably refers to animal tigers while it could refer to a football team if it appears in the sports news. On the other hand, two synonymous words may refer to the situation where different words could possibly have the same meaning, e.g. "excellent" and "terrific" can be used interchangeably. Therefore, matching only keywords might not capture the true semantic concepts due to the heterogeneity and independence of data sources and data repositories. On the other hand, the word-document matrix composed of document vectors is a very large matrix and it scales with the number of documents which could lead to disasters when the number of documents is huge.

LSI tries to statistically derive conceptual indices from a collection of documents (corpus) assuming there is an underlying latent structure in word usage through modeling the co-occurrence of keywords in documents. However, the performance of LSI based text representation heavily depends upon the quality of the corpus. On one hand, many document archives contain a large portion of short articles such as tweets, web blogs and posts, etc., which would lead to poor performance of LSI. On the other hands, tons of new words and phrases are being created benefiting from the fast spreading internet phenomenon. Without sufficient data, new words or phrases could not be captured by LSI.

Neural network based language models [Mikolov et al., 2013b, Le and Mikolov, 2014, Pennington et al., 2014, Collobert and Weston, 2008] try to represent words as dense, low-dimensional and real-valued vectors that can capture the semantic and

syntactic properties. These vectors, referred to as "word embeddings", are used abundantly by machine learning algorithms across a variety of tasks including text classification and clustering. Word2Vec [Mikolov et al., 2013b, Mikolov et al., 2013a] has recently gained a lot of interests in the text mining community recently. It maps words to low dimensional vectors to capture syntactic and semantic regularities. Although word embeddings can reduce the dimension of the word vectors, it fails to capture the long-distance semantic relations (semantic relations between words that are far apart from each other within the text) and the training set can be extremely large in order to achieve good performance.

Doc2Vec [Le and Mikolov, 2014] proposed two models , *Distributed Memory Model Paragraph Vectors (PV-DM)* and *Distributed Bag of Words Paragraph Vectors (PV-DBOW)*, trying to represent entire documents in a dense, low-dimensional space. *PV-DM* learns to predict the word using word and paragraph vectors while the paragraph vectors is directly learned to predict randomly sampled context words in *PV-DBOW*. Although word vectors capture semantics and are shared across different paragraphs of the document in both models, document vectors are learned over context words from the same paragraph and may capture only local semantics. Moreover, the training process to obtain paragraph vectors of Doc2Vec limits the application of Doc2Vec. For example, in the task of information retrieval, where there is a large demand to analyze new documents in real time, Doc2Vec models cannot be applied as it is impossible to get document vectors without re-training previous trained data.

All the above methods focus on exploiting the information within the text. They suffer the lack of domain knowledge as information from other sources cannot be integrated. We develop new methods to encode domain knowledge residing in WordNet.

4.2.1 WordNet

WordNet is a large lexical database of words where nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept with a variety of relations (Polysemous words will appear in one synset for each of its senses). The synsets are connected by semantic and lexical relations. Instead of being organized alphabetically as a standard dictionary, WordNet is organized conceptually. For instance, WordNet distinguish between the two senses of the noun "slot" with the synsets {shot, injection} and {short, snapshot}. The word "shot" in the sentence "The photographer took a shot of mine" can be replaced by "snapshot". While the "shot" in "The nurse gave me a flu shot" can be substituted with "injection", which is another sense of "shot".

WordNet (version 3.0) contains 15,5287 terms and 117,659 synsets, which are divided into four categories (11,7798 nouns, 11,529 verbs, 21,479 adjectives, 4,481 adverbs). Nouns, verbs, adjectives, and adverbs are organized differently in WordNet. Even though they are all organized in synset, the semantic relations among the synsets differ depending on the grammatical categories, as can be seen in Table 4.1

The *Hyponym/Hypernym* relationship and the *Meronym/Holonym* relationship are the most recognized relationships in WordNet. The hyponym of a noun is its subordinate, the relation between a hyponym and its hypernym is an *is a kind of* relation. For instance, "apple" is a hyponym of "fruit", which is to say that an "apple" is a kind of "fruit". Hypernymy (supername) and its inverse, hyponymy (subname), are transitive semantic relations between synsets. Meronymy (part-name), and its inverse holonymy (whole-name), semantic relations that distinguish component parts, substantive, and member parts. For example, "wheel" is a meronym of "car", which is to say that a "wheel" is part of "car". A fragment of the WordNet is illustrated

Semantic Relation	Syntactic Category	Examples
Synonymy (similar)	Noun	band, ring
	Verb	rise, ascend
	Adj	fast, quick
	Adv	readily, willingly
Hyponymy (subordinate)	Noun	apple, fruit
		tree, plant
		orange, color
Meronymy (part)	Noun	wheel, car
		ship, fleet
		leaf, tree
Antonymy (opposite)	Noun	top, bottom
	Verb	rise, fall
	Adj	fast, slow
	Adv	up, down

Table 4.1: Significant semantic relations in WordNet

in Figure 4.1. Unlike in the DAG graph of Gene Ontology where a node can have multiple parent nodes, in WordNet ontology, a term can only have one parent. The different structure of Gene Ontology and WordNet lead to different approaches of analyzing the semantic similarity between nodes.

The fact that synonyms are grouped in WordNet solves the synonmy problem to a great extent, it still suffers from polysemy as in other natural language processing problems. [Hotho et al., 2003] introduces a naive strategy regarding word concept as the most frequent used concept (synset ID) in the lexical database. Unfortunately, other meanings of the word may never be used due to this high bias on frequent word in the thesaurus even the document itself intends to.

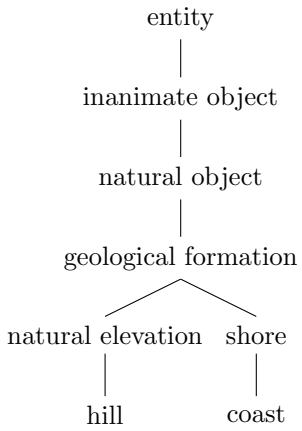

Figure 4.1: A fragment extracted from WordNet

4.3 Concept Tree Construction

We propose a *hybrid word embeddings (HWE)* model that can combine both text information as well as the pre-established structured knowledge in ontologies. The architecture of the proposed HWE system is shown in the upper part of Figure 4.2. We first extract concepts from the document by taking advantage of the knowledge in WordNet. These initial representations of the concepts are called "*Concept trees*" (Definition 1), which are constructed by linking the concepts in a document with relationship links found in WordNet. "Concept trees" will then be encoded into "tree vectors". The details are shown as follows.

Definition 1. ***Concept tree***. A concept tree consists of concepts within a text document that are connected with relationships obtained from ontologies.

The construction of concept trees consists of 3 steps: stop-word removal, stemming and linking.

Stop-word removal: Stop words, words that do not contribute much to the semantic meaning such as "the", "a", "is", "which", are removed from the text.

Stemming: Words with the same root appear in various morphological forms. To re-

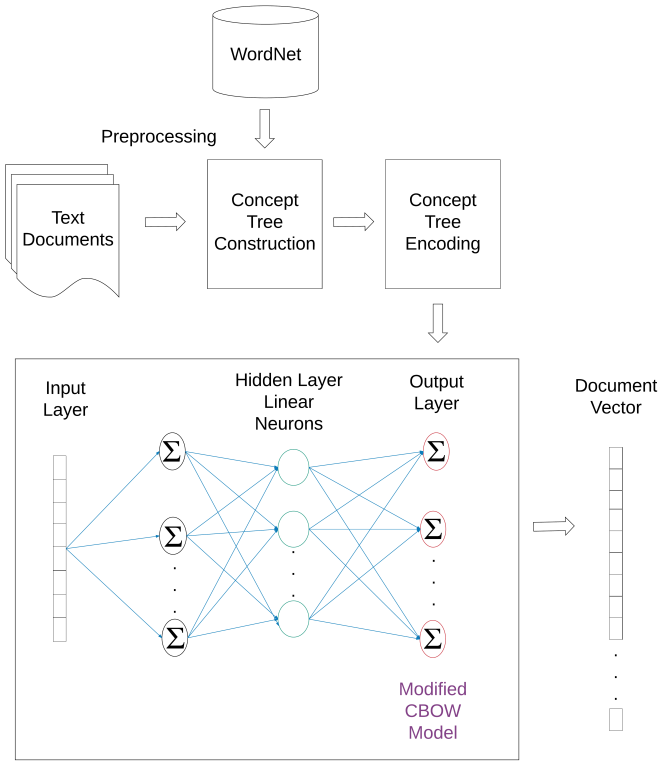

Figure 4.2: The architecture of the semantic system incorporating domain knowledge

duce the noise when applying similarity metric to the text, all the words are stemmed using the morphology function provided with WordNet. For instance, "takes", "took" and "taking" are all mapped to the root word "take". Since we do not consider the word ordering, the word "taken" can also be mapped to "take" ignoring the subject. After stemming, each word obtained corresponds to a synset ID, which constitutes a set of synonyms.

Linking: Terms obtained from the same text document after preprocessing are connected via a hypernymy relationship between their associated synset IDs, if there is any. A hypernym of a term is a more general term and a hyponym is a more specific term. For example, an apple is a hyponym of edible fruit and an edible fruit is a hypernym of an apple. This hypernym relationship from WordNet is exploited to build the concept trees. For example, an article studying whether eating an apple a day can prevent from cancer might contain words like "food", "fruit", "apple", "tumor", "cancer". We thus can construct a concept tree for terms "food", "fruit", "apple" and "tumor" and "cancer" separately as shown in Figure 4.4. As words are expressed as synsets in WordNet and a word may have multiple meanings, i.e., multiple synsets. All the synsets of a word are used in concept tree construction. For each synset, the associated hypernym trees are obtained through the hypernym relationship. The hypernym trees contain all the ancestors of the synset. Each synset contains multiple words that have the same meaning, thus a hash table is built with words as indexes and words associated with each synset in the hypernym trees are treated as values. We scan the article twice. The first scan builds the hypernym hash table. The second scan constructs the concept trees.

A word in different contexts may represent different concepts. For instance, the word "orange" could be interpreted as a color as in "The sweater is orange.". It could also be regarded as an editable fruit as in "The orange is so delicious!". Thus,

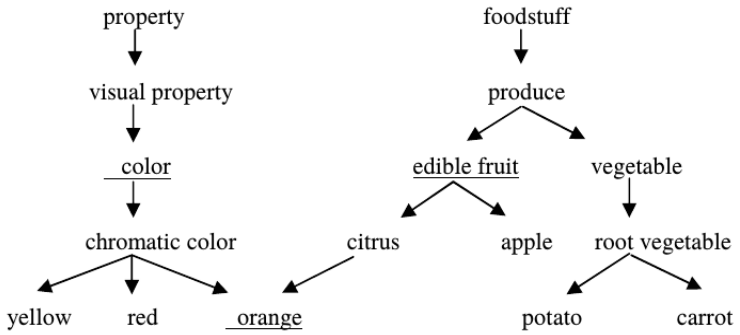

Figure 4.3: Hypernym trees of the term "orange"

a word may be placed in different synsets, which form different concept trees. For example, to look up the hypernym of the word "orange" with the color concept , the left hypernym tree in Figure 4.3 should be followed; the "orange" with the fruit concept should lead to the hypernym tree in the right.

It is hard to determine the correct concept for an ambiguous word from several synsets and so is deciding the semantics of a document that contains several ambiguous terms. Wang [Wang and Taylor, 2007] directly uses the first synset of a word which has the highest frequency of occurrence in WordNet as the word concept. Using this approach, all ambiguous words are represented by the same concept. For instance, the first synset of "bank"in WordNet represents the sloping land (especially the slope beside a body of water). When it comes to finance news, the synset of "bank" denotes a financial institution that accepts deposits and channels the money into lending activities. To avoid mismatching concept, we will keep all the synsets and build individual concept trees for different concepts of ambiguous words (words that have multiple synsets in WordNet). We will then use the text context to assist the disambiguation during the HWE process.

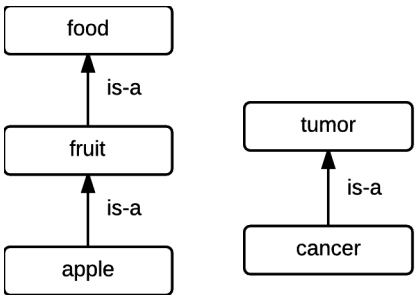

Figure 4.4: Concept trees derived from terms "food", "fruit", "apple", "tumor", "cancer"

4.4 Concept Tree Encoding

In this section, we incorporate domain knowledge from WordNet as well as word statistics into concept trees. We first define several terms that will be used in the encoding phase.

Definition 2. ***Semantic content value***. The sum of frequencies of words mapped to a synset ID i is the semantic content value of synset ID i, denoted as ω_i. The sum of the semantic content values of all the synset ID associated with a concept k is the semantic content value of the concept k, denoted as SCV_k.

$$SCV_k = \sum_{i \in T_k}^{n} \omega_i \tag{4.1}$$

where T_K is the set of words constituting concept k.

Definition 3. ***Semantic content rate (SCR)***. The semantic content value of a concept k normalized by the total sum of the semantic content value of all the concept in a concept tree is the SCR of the concept k, which is defined as,

$$SCR_k = SCV_k / \sum_{i=1}^{n} SCV_i \tag{4.2}$$

However, words at upper layers of WordNet have more general concepts and less semantic similarities between words than words at lower layers. This implies more general terms convey less information than more specific terms at lower levels. For instance, an article discussing "the impact of eating apple on avoiding cancer" may probably mention the word "fruit", which would indicate the 'fruit-apple' hypernymy relation constructed in a concept tree. High frequencies of "fruit" would mislead the topic to represent the concept "fruit" instead of "apple". Key points of the text might not be captured as the word "fruit" also applies to other fruits like "pineapple". This could affect the classification precision or the recall. For example, in information retrieval scenario where the search query contains "apple", the above article that has both "fruit" and "apple" might have a rather low rank even though the article is talking about the "apple" instead of the more general concept "fruit". Hence, the impact of words at upper levels should be dampened while lower level words should have higher impact since they convey more unique information than the upper level ones.

In [Song et al., 2014], statistical approaches were used to address the semantic impacts of different ontological levels. However, this is impractical for short text documents, as words are not likely to appear many times as they would in longer documents, which would result in a sparse term frequency matrix that can hardly capture the underlying structure of the document.

The scaling depth effect in [Li et al., 2003] exhibits a promising way to model the impact of different levels on the WordNet. This effect $\theta(h)$ of level h is captured by,

$$\theta(h) = \frac{1}{1 + e^{-\gamma h}} \tag{4.3}$$

We note that $\theta(h)$ is a monotonically increasing function with respect to depth h,

while γ acts as an scaling factor that can be used to control the extent of the depth effect. As a result, it would scale up the impact of word frequencies at lower layers while scaling down those at upper layers. By adjusting the contribution of depth of each synset ID, we have the following definitions,

Definition 4. ***Weighted semantic content value***. The sum of the semantic content values of all the synset ID associated with a concept k weighted by a scaling depth effect $\theta(h)$ is the weighted semantic content value of the concept, denoted as $WSCV_k$,

$$WSCV_k = \sum_{h=1} \phi(h) SCV_k \tag{4.4}$$

Definition 5. ***Weighted Semantic content rate (WSCR)***. The weighted semantic content value of a concept k normalized by the total weighted sum of the semantic content value of all the concepts within a concept tree is the WSCR of the concept k defined as,

$$WSCR_k = WSCV_k / \sum_{i=1}^{n} WSCV_i \tag{4.5}$$

We use V dimension vectors to encode concept trees, where V is the vocabulary size. The vocabulary is typically defined using the V most common words in the text corpus. To encode concept tree, we define tree vectors as follows,

Definition 6. ***Tree vector***. Vector with weighted semantic content rate at each dimension representing concepts in a concept tree.

Taking the "tumor-cancer" shown in Figure 4.4 as an example, suppose $ct_{tumor} = 10, ct_{cancer} = 5$, γ in Equation 4.3 is 0.5. The tree vector of "tumor-cancer" will have $\frac{10/(1+e^{-0.5*0})}{10/(1+e^{-0.5*0})+5/(1+e^{-0.5*1})} = 0.62, \frac{5/(1+e^{-0.5*1})}{10/(1+e^{-0.5*0})+5/(1+e^{-0.5*1})} = 0.38$ at the dimension of

"tumor" and "cancer", respectively and 0 elsewhere. The tree vectors generated are the target values representing the context. The procedure is shown in the next section.

4.5 Hybrid Word Embeddings (HWE)

Distributed word representations (word embeddings) were explored in the NLP community [Mikolov et al., 2013a, Blei, 2012, Le and Mikolov, 2014] where words are represented as dense, low-dimensional vectors. Each dimension is a measure of the association between the word and a particular context. To incorporate WordNet knowledge to enrich semantics of text content, we propose a neural network model to generate word embeddings that not only contain the context information, but also the knowledge extracted from WordNet.

Our proposed hybrid word embeddings model is adapted from the continuous bag-of-word model (CBOW) [Mikolov et al., 2013a] by re-engineering the outputs and the objective function. In our setting, the vocabulary size is V, and the hidden layer size is N. The nodes between adjacent layers are fully connected. We first introduce the model with only one word in the context and then extends to the scenario where multiple words are in the context. If there is only one word in the context, the input would be a one-hot vector and the weights between the input and output layer can be represented by a $V \times N$ matrix $\mathbf{W}$, shown in Figure 4.5. Each row of $\mathbf{W}$ is the N-dimension vector representation v_w of the associated word of the input layer.

$$\mathbf{h} = \mathbf{W^T x} = \mathbf{v^T_{w_I}} \tag{4.6}$$

Given a context (a single word in this case), assuming $x_k = 1$ and $x_{k'}$ for $k' \neq k$, we are essentially copying the $k - th$ row of W to h. $\mathbf{v_{w_I}}$ is the vector

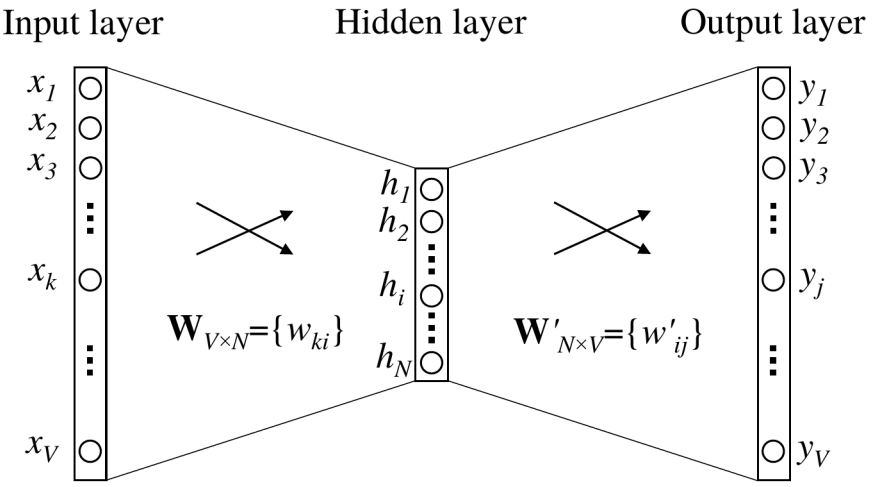

Figure 4.5: Graphical representation of the proposed HWE model with only one word in the context

representation of the input word w_I. The activation function of the hidden layer is linear (i.e., directly passing its weighted sum of inputs to the next layer). There is a different $N \times V$ matrix $\mathbf{W}'$ from the hidden layer to the output layer. Using these weights, a score u_j for each word in the vocabulary is computed as,

$$u_j = \mathbf{v}'_{\mathbf{w_j}}\mathbf{h} \tag{4.7}$$

where $\mathbf{v}'_{\mathbf{w_j}}$ is the j-th column of the matrix $\mathbf{W}'$. We further use softmax to obtain the posterior distribution of words, mapping the values of $\mathbf{v}'_{\mathbf{w_j}}$ to the range 0 to 1.

$$p(w_j|w_I) = y_j = \frac{exp(u_j)}{\sum_{i=1}^{V} exp(u_i)} \tag{4.8}$$

where y_j is the output of the jth dimension in the output layer. Combining Equation 4.6 and 4.7, we get

$$p(w_j|w_I) = \frac{exp(\mathbf{v}'^{\mathbf{T}}_{\mathbf{w_j}}\mathbf{v}_{\mathbf{w_I}})}{\sum_{i=1}^{V} exp(\mathbf{v}'^{\mathbf{T}}_{\mathbf{w_i}}\mathbf{v}_{\mathbf{w_I}})} \tag{4.9}$$

Unlike the CBOW model which uses the target word as the output, we instead use tree vectors derived in the above sections as the output. For each word, its corresponding tree vector is directly regarded as the output. Instead of predicting a single word, we use tree vectors to connect all the related words. Tree vectors can capture more semantics and require smaller training corpus with the help of information residing in existing knowledge bases. For example, the two sentences "I love eating apple", "I love eating fruits" are both valid where the word "apple" and the word "fruits" are related. However, a traditional CBOW model treats "apple" and "fruits" as totally unrelated individual words, even though human can easily tell "apple" is a kind of "fruit" using common sense. The relation between "apple" and "fruit" is described as a hypernymy relationship in WordNet. As the two words share a similar context, with the help of our encoded tree vectors, training the word "apple" in the sentence "I love eating apple" will also train the word "fruit" in "I love eating fruit" as "fruit" and "apple" are connected in the encoded tree vector.

Therefore, we predict the tree vector which might contain multiple target values in the output layer. We use ***Mutilabel Soft Margin Loss*** as the object function to predict multiple targets.

$$loss(x, y) = -\sum_{i=1}^{V} \frac{t[i] \log \frac{e^{y[i]}}{1+e^{y[i]}} + (1 - t[i]) \log \frac{1}{1+e^{y[i]}}}{V} \tag{4.10}$$

where $t[i]$ is the the i-th element in the tree vector and $y[i]$ is the i-th element of the output, shown in Equation 4.8. The training objective (for one training sample) is to minimize the loss in Equation 4.10. The weights are updated by stochastic gradient descent and backpropagation. According to stochastic gradient descent, the corresponding weights are updated with a proportion to the $loss$ in Equation 4.10. If the predicted output has a large difference ($loss$) from the target tree vector, the

corresponding weights will be changed significantly. Conversely, if the output vector is close to the target tree vector (small loss), meaning word (concept) may be fairly accurately predicted, it will have little effect on the corresponding weights. The final $V \times N$ $\mathbf{W}$ matrix can be viewed as the word representation matrix, where each word is represented as an N-dimension vector. As the word vectors generated from the HWE model contain context information as well as encoded knowledge from WordNet (tree vectors), we call these word vectors *hybrid word embeddings*.

We now show the model with a multi-word context setting. When computing the hidden layer output, instead of copying the input vector of the input context word, the model takes the average of the input context word vectors, and use the product of of the average vector and the weight matrix $\mathbf{W}$ as the output,

$$\mathbf{h} = \frac{1}{C}\mathbf{W^T}(\mathbf{x_1} + \mathbf{x_2} + \cdots + \mathbf{x_C}) \tag{4.11}$$

where C is the number of words in the context. $\mathbf{x_1}, \cdots, \mathbf{x_C}$ are the corresponding input word vectors. The objective function stays the same. An example of the model with a context of three words is shown in Figure 4.6. Both Figure 4.5 and 4.6 are taken from [Rong, 2014].

4.6 Document Vector

The hybrid word embeddings obtained from the proposed HWE model can be regarded as new representations of words. They are vectors of a fixed dimension. Here, we further extend our word embeddings to document representations. A document is a collection of words, where similar words convey semantics at similar dimension in the word embeddings while the dimensions expressed by dissimilar words may have

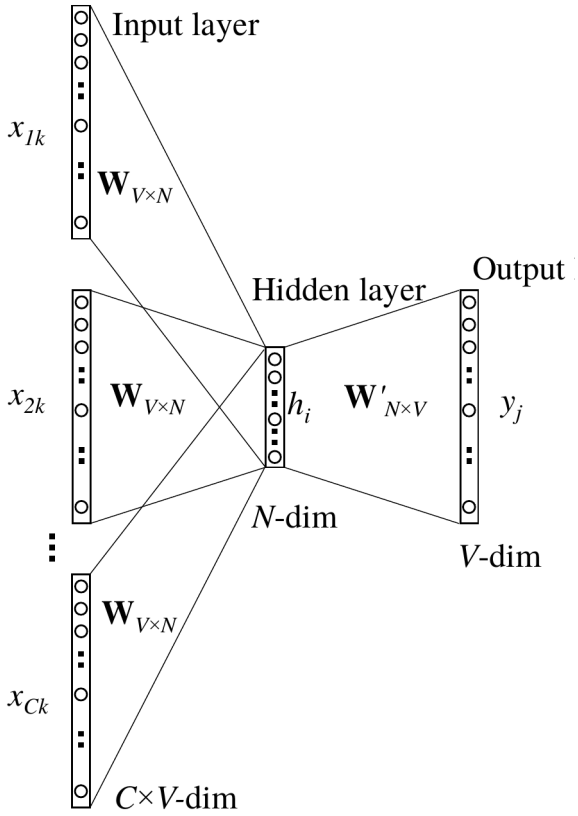

Figure 4.6: Graphical representation of the proposed HWE model with three words in the context

less overlaps, if at all. Based on this observation, we adopt a simple **Normalized sum** strategy to generate document vectors.

Normalized sum. Word embeddings express semantics through the values at each dimension. These semantics consist of information from both the text context and knowledge from WordNet. For each document, we sum up all the word embeddings to obtain a document vector. The summation is then normalized by the total number of words. If a word appears multiple times in a document, their corresponding word embeddings will also be included in the summation the same number of times. The final document vector is a single vector of the same dimension as the word embeddings generated from HWE. The similarity between two documents is computed as the cosine similarity of the corresponding two document vectors. Once we trained the hybrid word embeddings from the proposed HWE model, each document vector can be formed in linear time with respect to the length of the document. Compared to Doc2Vec [Le and Mikolov, 2014] where document vectors need to be inferred from the model or trained through the entire corpus, the **Normalized sum** method is more flexible as it constructs document vectors directly from pre-trained word embeddings.

4.7 Performance study of HWE

Text classification is one of the most important tasks in natural language processing and information retrieval due to the increased availability of documents in digital form and the ensuing need to access them in flexible ways. It is the activity of labeling natural language texts with thematic categories from a pre-defined set. By assigning documents to labeled classes, text classification can reduce the search space and expedite the process of retrieving relevant documents.

We investigate the effectiveness of the proposed HWE model in terms of multi-class and multi-label document classification. In this experiment, we compare HWE with the most popular approaches, including Bag-of-Words (BoW) model [Harris, 1954], Latent Semantics Indexing (LSI) [LANDAUER and DUMAIS, 1997], Doc2Vec [Le and Mikolov, 2014] (PV-DM, PV-DBOW), Word2Vec [Mikolov et al., 2013a], and Google News pre-trained Word2Vec. The Google News pre-trained word embeddings are obtained from (`https://code.google.com/archive/p/word2vec/`) which contains 300-dimensional vectors for 3 million words and phrases. These embeddings were trained by the skip-gram model. For those state-of-the-art models, the best parameter settings reported by the respective papers are used. We use 400 dimensions for paragraph vector model, 300 for Word2Vec models and 300 for our model. The experiment is conducted on a Linux box with a i7-2600K CPU @ 3.40GHz, 32G memory, Nvidia GTX 1070 8GB graphics card. The preprocessing of documents and generation of "tree vectors" is done by the Python NLTK package [Loper and Bird, 2002], the neural network model is implemented by the PyTorch package [pyt, 2017]. The other popular methods in this experiment are implemented with the gensim package [Řehůřek and Sojka, 2010].

4.7.1 K-Nearest Neighbors

We adopt k-Nearest Neighbors (k-NN) based classification as the classification method. The principle behind k-NN is to find k training samples closest in distance to the new point, and predict the label from these. In terms of document classification, k-NN classifies an unseen document to its k nearest neighbors in a specified training set. The classification result is computed from the majority vote of the k nearest neighbors of each document: a query document is assigned the document label which has the most representatives within the nearest neighbors of the document. The optimal choice of the value k is highly data-dependent: When k is small, the region of the prediction is restrained. A small value of k provides the most flexible fit, which will have low bias but high variance. On the other hand, a higher k averages more voters in each prediction and is more resilient to outliers. Larger values of k suppresses the effects of noise, but makes the classification boundaries less distinct. For each document d, we compute the set D_k containing the k most similar documents to d with corresponding label set L_k. Then d is classified to class c which appears most frequently in L_k.

The best k in k-NN model is the one that corresponds to the lowest test error rate. If we carry out repeated measurements of the test error for different values of k in the test set, we are actually using the test set as a training set. This means we are underestimating the true error rate since our model has been forced to fit the test set in the best possible manner. In this case, the model is incapable of generalizing to newer observations, a.k.a. overfitting.

To solve the above issue, we can estimate the test error rate by holding out a subset of the training set. This subset, called the validation set, can be used to select the appropriate level of flexibility of the model. We explore K-fold cross validation

to get the best k. Note that K in K-fold cross validation is totally unrelated to k as in k-NN. K-fold cross validation involves randomly dividing the training set into K groups (folds) of approximately equal size. We treat one of the K folds as the validation set, and the remaining $K-1$ folds as the training set. This procedure is repeated K times, with each of the K subsamples used exactly once as the validation set. The above process results in K estimates of the test error which are then averaged out. In this experiment, k in both baselines and the proposed model are tuned by 10-fold cross validation.

4.7.2 Data sets

Data sets: Two data sets 20 newsgroups and Reuters-21578 are used in the evaluation.

- 20 newsgroups. It is a collection of 19,974 newsgroup documents, partitioned evenly across 20 different newsgroups. Some of the news groups are very closely related to each other (e.g., comp.sys.ibm.pc.hardware / comp.sys.mac.hardware), while others are highly unrelated (e.g., rec/autos / talk.politics.guns). We randomly select 70% of the documents to be used for training and the remaining 30% for testing.

- Reuters-21578. It is commonly used for text classification during the last two decades. It contains thousands of documents collected from Reuters newswire in 1987. This collection consists of 21,578 documents, including topics and typographical errors. A subset and split of the collection, referred to as *ModApte*, is used. This split assigns documents before April 7, 1987 to the training set, and documents after April 8, 1987 to the test set. An additional step is to focus only on the categories that have at least one document in the training set and

the test set. After this, the data set has 90 categories with a training set of 7796 documents and a test set of 3019 documents.

4.7.3 Multi-class classification

Multi-class classification refers to classification task with more than two classes. We evaluate the performance of our proposed document similarity measurement using multi-class classification.

We evaluate the performance using the following metrics, macro-average classification ***Recall***, ***Precision***, ***Accuracy***, ***F1-Score*** which are defined as follows, The parameters of the above measurements are summarized in Table 4.2.

- **Recall$_i$**. The ratio of documents classified to the class c_i within the total documents in the class c_i.

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \tag{4.12}$$

- **Precision$_i$**. The ratio of documents classified correctly in the class c_i within the documents assigned to the class c_i.

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \tag{4.13}$$

- **Accuracy$_i$**. The ratio of documents classified correctly in the class c_i and not in class c_i within the total number of documents.

$$Accuracy_i = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \tag{4.14}$$

- **F1** *Score*. The harmonic mean of precision and recall.

$$F1_i = \frac{2Precision_i Recall_i}{Precision_i + Recall_i} \tag{4.15}$$

The macro-average precision, recall, accuracy are taken as the average of the precision, recall, accuracy on different classes. The macro-average F1-Score is the harmonic mean of the macro-average precision and recall.

	Belong to c_i	Not belong to c_i
Classified to c_i	TP_i	FP_i
Not classified to c_i	FN_i	TN_i

Table 4.2: Contingency Table

- TP_i: number of correctly classified documents as in c_i, which belong to the class c_i.
- FP_i: number of incorrectly classified documents as in c_i, which do not belong to the class c_i.
- FN_i: number of incorrectly classified documents as not in c_i, which belong to the class c_i.
- TP_i: number of correctly classified documents as not in c_i, which do not belong to the class c_i.

Table 4.3 shows experimental results of all models we examined on the 20 newsgroups data set. The proposed HWE model outperforms all the other models in all aspects. The major problem of Doc2Vec (PV-DM, PV-DBOW) is that the document vectors need to be trained by the entire corpus at the cost of time efficiency and computation resource. The extensive labor and procedure to train documents to

get document embeddings inevitably limit the application of Doc2Vec. Due to the large amount of documents, it is hard to train Doc2Vec models in real time to obtain document embeddings. The rapid generation of new online documents only aggravate the situation. New blogs or tweets cannot be identified, classified or even analyzed in real time using the Doc2Vec model. The proposed HWE model can construct document representations in linear time in proportion to the lengths of documents. As the semantic relationships between concepts are already extracted in the training process of generating hybrid word embeddings, when constructing document vectors, we only need to scan the document once to apply the *Normalized sum* to the words composing the documents. In the scenario where a document contains new words that have not appeared in the training set, existing methods including Word2Vec and Doc2Vec simply ignore the words and hence the semantics of the words cannot be captured. If the new words can be found in WordNet, the proposed HWE model can construct concept trees of the new words by linking related concepts in WordNet. Depending on the application, the new words can be attributed to the root or the combinations of all the nodes along the path to the root in corresponding concept trees. The proposed HWE model becomes rather useful when training data is not sufficient. In the case where the new words are not recorded by WordNet, the proposed HWE model can capture the context information by re-training the entire corpus with the new document, as Word2Vec or Doc2Vec can do. Pre-trained Word2Vec model from Google News performs better than the Word2Vec model trained on the 20 newsgroups data set as the pre-trained Word2Vec was trained by a much larger corpus. HWE incorporates domain knowledge from WordNet into word embeddings to enrich text content expressed by word embeddings. The proposed HWE model can use smaller training corpus to get better results.

Model	Accuracy	Recall	Precision	F1-Score
HWE	**0.761**	**0.833**	0.857	**0.845**
$PV-DM$	0.744	0.818	0.844	0.831
$PV-DBOW$	0.722	0.793	0.852	0.821
$Word2Vec$	0.621	0.667	0.833	0.740
$Word2Vec_{pre}$	0.667	0.714	**0.862**	0.781
LSI	0.648	0.75	0.778	0.764
BOW	0.535	0.588	0.769	0.667

Table 4.3: Performance of multi-class classification on 20 newsgroups

4.7.4 Multi-label classification

Multi-label classification assigns to each sample a set of target labels. In the case of document classification, a document (sample) can have multiple topics (categories). For example, a document might be about finance, sports, education at the same time. The performance of the proposed model is evaluated by multi-label classification on the Reuters data set where documents can have multiple labels (categories). In multi-class classification problem, the above Precision, Recall, F1-Score standard evaluation metrics are usually defined. However, in multi-label classification, predictions are a set of labels and the prediction can be fully correct, partially correct or fully incorrect. None of these existing evaluation metrics capture such notion in their original forms. We will explore the following metrics, ***Coverage Error*** and ***Label ranking average precision score (LRAPS)***. We use one-vs-all settings to train a single classifier per class, with the documents of that class as positive samples and all other documents as negatives.

- **CoverageError**. The average number of labels that are included in the final prediction such that all true bales are predicted. The best value of this metrics the average number of true labels. Formally, given a binary indicator matrix of the ground truth labels $y \in \{0,1\}^{n_{samples} \times n_{labels}}$ and the predicted score associ-

ated with each label $\hat{f} \in \mathbb{R}^{n_{samples} \times n_{labels}}$, the coverage is defined as,

$$coverage(y, \hat{f}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \max_{j:y_{ij}=1} rank_{ij} \tag{4.16}$$

with $rank_{ij} = |\{k : \hat{f_{ik}} >= \hat{f_{ij}}\}|$.

- **LRAP**. The average of the ratio of true against total labels with lower score. The metrics will yield higher scores if labels associated with each sample are given better ranks. The value of LRAP is greater than 0, and the best value is 1. Formally, given a binary indicator matrix of the ground truth labels$y \in \{0, 1\}^{n_{samples} \times n_{labels}}$ and the predicted score associated with each label $\hat{f} \in \mathbb{R}^{n_{samples} \times n_{labels}}$, LRAP is defined as,

$$LRAP(y, \hat{f}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \frac{1}{|y_i|} \sum_{j:y_{ij}=1} \frac{|\mathcal{L}_{ij}|}{rank_{ij}} \tag{4.17}$$

with $\mathcal{L}_{ij} = |\{k : y_{ik} = 1, \hat{f_{ik}} >= \hat{f_{ij}}\}|$ and $rank_{ij} = |\{k : \hat{f_{ik}} >= \hat{f_{ij}}\}|$.

Table 4.4 reports the comparison of the proposed hybrid model against baselines. The pre-trained Word2Vec model still performs better than the Word2Vec model trained using the Reuters data set because the pre-trained Word2Vec word embeddings was trained by a much larger data set which covers a variety of topics. Hence, pre-trained Word2Vec word embeddings are more powerful and can capture more semantics than the embeddings trained on the Reuters data set. PV-DM and PV-DBOW outperform all the other methods as Doc2Vec was specially designed to train document embeddings to represent documents incorporating the topic factor which are useful in multiple topics related NLP tasks such as multi-label classification. However, the labor and procedure to train documents to get document embeddings

inevitably limits the application of Doc2Vec. The rapid generation of new online documents only aggravate the situation. New blogs or tweets cannot be identified, classified or even analyzed in time using the Doc2Vec model.

Model	Coverage Error	LRAPS
HWE	**10.51**	**0.925**
$PV-DM$	11.18	0.918
$PV-DBOW$	10.69	0.920
$Word2Vec$	12.17	0.891
$Word2Vec_{pre}$	11.55	0.910
LSI	18.49	0.824
BOW	19.67	0.799

Table 4.4: Performance of multi-label classification on Reuters-21578

4.7.5 Dealing with Unseen Words

One of the major issues of traditional document representation models is that they cannot deal with words that have not appeared in the training set before (unseen words). Models like Word2Vec and Doc2Vec have to either discard the new words or re-training the entire model with the added new words. The proposed HWE model can quickly extend to accept unseen words with the help of ontology or the context information. We consider two scenarios: (1) The unseen words can be found in ontology such as WordNet. Then the words can be attributed to their related concepts in the ontology. (2) The unseen words cannot be found in the ontology. Then the words can be attributed to their context information in the text. There are two cases in each scenario whether we need to re-train the entire model or not.

In the first scenario, we use the related concepts of the unseen words in WordNet to represent the unseen words. We take the average of the word embeddings of its synonyms and ancestors in the WordNet, if at all, as the word embeddings of the unseen words. For example, if the word "band" is the unseen word that has never

appeared before while "ring" and "jewelry" showed up in the training corpus, we look up in the WordNet and realize that "ring" is a synonym of "band" and "jewelry" is a hypernym of "band" when "band" is referred to as a kind of jewelry consisting of metal worn on the finger. We hence take the average of the word embeddings of "ring" and "jewelry" that have already been trained in the model before as the new word embeddings of the unseen word "band". We call the word embeddings obtained in this way *Accumulate Word Embeddings From Ontology (AWE-O)*. Using AWE-O, we could gain word embeddings of the unseen words while models like Word2Vec cannot. On the other hand, "band" could also refer to a group of instrumentalists and the word "music" might likely appear in the context of "band". If we have sufficient context information, semantics of words related to the surrounding text can also be captured. In this case, we perform a re-training of our HWE model to capture the context information. As we can relate the unseen words to their related concepts in WordNet, the target *tree vector* will stay the same. Even though there may not be sufficient context information of the unseen words, the semantics of their related concepts captured by HWE can be attribute to the unseen words as well.

In order to test the ability of predicting unseen words of the proposed HWE model, we divide the data set into two parts where each document in one part contains at least one unseen word and the other part do not contain unseen words at all. We further divide the set of document that unseen words appear in into two equal parts, and take one part along with the set of documents which do not have unseen words as the training set. The other part of the documents containing unseen words is treated as the test set. After stemming, there are 100373 distinct words in the 20 newsgroup and WordNet has recorded 147306 distinct words in total. There are 13498 words that WordNet and the 20 newsgroup have in common. To achieve this, we first build inverted index of the common words that appear in both the WordNet and the data

set. We randomly select one third of these words as the unseen words. We take the union of the set of documents that contain the unseen words and randomly select one part as the test set, while the other part combining with the set of documents that do not contain the unseen words are treated as the training set. To reduce the bias on the random selection of words and set of documents, we repeat the experiment 10 times and take the average as the result.

The result is shown in Table 4.5 when we use HWE-O to attribute the unseen words to its synonyms and hypernyms in the WordNet. The proposed HWE model still outperforms all the other models. We can also see that the fact that Word2Vec performs worse than LSI and BOW in this case indicates the performance of Word2Vec models highly rely on the quality of the data set. Table 4.6 shows the performance evaluation when we re-train the entire model to take the context information into consideration. The entire performance of each model is better than the one directly using AWE-O without re-training as the model captures the context information of the unseen words.

Model	**Accuracy**	**Recall**	**Precision**	**F1-Score**
HWE	**0.690**	**0.792**	**0.826**	**0.809**
$PV-DM$	0.615	0.750	0.724	0.737
$PV-DBOW$	0.658	0.759	0.786	0.772
$Word2Vec$	0.520	0.588	0.667	0.625
$Word2Vec_{pre}$	0.559	0.696	0.667	0.681
LSI	0.571	0.650	0.722	0.684
BOW	0.535	0.588	0.769	0.667

Table 4.5: Performance of multi-class classification on 20 newsgroups with unseen words that do not appear in the data set before but in WordNet without re-training

It is possible that the related concepts of the unseen words in WordNet do not contain the true meanings of the unseen words in the text, which means that the unseen words have new senses that have no corresponding related entries in WordNet.

Model	Accuracy	Recall	Precision	F1-Score
HWE	**0.778**	**0.818**	**0.900**	**0.857**
$PV-DM$	0.718	0.781	0.862	0.820
$PV-DBOW$	0.684	0.767	0.821	0.793
$Word2Vec$	0.609	0.714	0.758	0.735
$Word2Vec_{pre}$	0.630	0.722	0.788	0.754
LSI	0.617	0.722	0.765	0.743
BOW	0.581	0.688	0.733	0.710

Table 4.6: Performance of multi-class classification on 20 newsgroups with unseen words that do not appear in the data set before but in WordNet with re-training

For example, the word "bands" can also mean "one thousand dollars". There are also cases where unseen words that appear in the document do not belong to WordNet, especially when it comes to abbreviations or internet words that have not been recorded by WordNet. We call these unseen words and words with senses that appear neither in the training set nor the WordNet *new words*. For example, if "DHR" has not appeared in any documents before, tradition models would simply discard the term. While there are no corresponding entries in WordNet, the document where the new word ("DHR") appears should tell use the meaning of the word. For instance, "DHR" may actually refer to the company "Danaher Corporation". "DHR" appears in a finance article which is the company's stock report. Words like "share", "stock", "financial" would very likely to appear in the report. Instead of discarding the term, we train our models based on two observations: (1) The context information still tells us about the new word. "DHR" may possibly come directly after "Danaher Corporation" and thus we know "DHR" is a corporation. (2) The surrounding context may not tell us the whole picture about the new word. In fact, "Danaher Corporation" is a conglomerate in fields of design, manufacturing and healthcare. The entire article may be related to the "DHR". We should also exploit the global information (all the words appeared in the document) of the article. Based on the above observations, we

adopt a hybrid approach that tries to compensate for both aspects. If the model is not re-trained, we will take the *document vector* of the article as the word embeddings of the new word, i.e, the new word shares the same representation with the document it appears. If there are multiple documents containing the new word, the average of the document vectors is taken as the word embeddings of the new word. We call the word embeddings obtained in this way *Accumulate Word Embeddings From Text (AWE-T)*. The more documents containing new words, the more semantics the AWE-T can capture. As we have a certain amount of documents containing new words, the noise words in other parts of the documents may mislead the semantics of the new words. In this case, we need to focus on the surrounding context of the new words. We perform a re-training of our HWE model with slight differences. As there are no corresponding entries for the new words in WordNet, the only non-zero element in the *tree vector* is the word itself, the same as Word2Vec and Doc2Vec. If the number of documents containing the new words is relatively small, there would not be enough semantics to capture solely based on the surrounding context. Instead of randomizing the entire initial $\mathbf{W}$, we initialize the corresponding entries of new words in matrix $\mathbf{W}$ with the AWE-T of the new words. As the global information of the new words is already stored in AWE-T, our model can achieve better results.

To evaluate the ability of adapting to new words of the proposed HWE model, we use the same method to acquire both the training set and test set as the one in the first scenario. Table 4.7 and Table 4.8 show the performance of HWE compared with other models in the second scenario where new words cannot be found the previous documents nor the WordNet. The performance of other models stay the same as in Table 4.5 and Table 4.6, while the proposed HWE with AWE-T still outperforms other models. From Table 4.5-4.8, we observe that after re-training, HWE performs better than using AWE-T without re-training as the context information is captured,

yet it performs worse than the AWE-O method. This is because rather than using the whole text information to deduce the semantics of the new words, AWE-O precisely targets related concepts using WordNet, effectively filtering out noises (irrelevant concepts).

Model	**Accuracy**	**Recall**	**Precision**	**F1-Score**
HWE	**0.664**	**0.763**	**0.818**	**0.789**
$PV-DM$	0.615	0.75	0.724	0.737
$PV-DBOW$	0.658	0.759	0.786	0.772
$Word2Vec$	0.520	0.588	0.667	0.625
$Word2Vec_{pre}$	0.559	0.696	0.667	0.681
LSI	0.571	0.650	0.722	0.684
BOW	0.535	0.588	0.769	0.667

Table 4.7: Performance of multi-class classification on 20 newsgroups with new words that do not appear in the data set before nor in WordNet without re-training

Model	**Accuracy**	**Recall**	**Precision**	**F1-Score**
HWE	**0.725**	**0.788**	**0.867**	**0.825**
$PV-DM$	0.718	0.781	0.862	0.820
$PV-DBOW$	0.684	0.767	0.821	0.793
$Word2Vec$	0.609	0.714	0.758	0.735
$Word2Vec_{pre}$	0.630	0.722	0.788	0.754
LSI	0.617	0.722	0.765	0.743
BOW	0.581	0.688	0.733	0.710

Table 4.8: Performance of multi-class classification on 20 newsgroups with new words that do not appear in the data set before nor in WordNet with re-training

Chapter 5

Conclusions

In this dissertation, we developed a series of techniques to measure the semantic similarity of objects in multiple domains. By utilizing the structured knowledge that has already been established, we explore the domain knowledge from the existing lexical resources and incorporate it into specific applications within different domains.

In biology domain, we proposed a novel aggregate information content (AIC) method to measure the semantic similarity of GO terms accurately and efficiently using Gene Ontology. This AIC approach aggregates the information content of all ancestor terms of a particular GO term while the computation of GO term's information content implicitly considers the semantic contribution of its descendant terms. Thus, this approach ensures the completeness of the semantic information in the semantic similarity measure. Our analysis and experimental results show the superiority of the proposed AIC method over the state-of-the-art methods [Resnik, 1999, Lin, 1998, Jiang and Conrath, 1997, Wang et al., 2007]. We further enhance the popular G-SESAME Website [Wang et al., 2007] `http://bioinformatics.clemson.edu/G-SESAME` by providing Web services for GO term semantic similarity measure and gene functional similarity measure using different methods, including the proposed

AIC method, [Resnik, 1999], [Lin, 1998] and [Jiang and Conrath, 1997] methods respectively.

In text domain, we proposed the Hybrid Word Embeddings (HWE) model which combines semantic information obtained from WordNet and context information extracted from text documents to provide concise and accurate representations of text documents. Experimental study on classification of documents shows that HWE outperforms the popular methods, Vector Space Model (VSM) model [Harris, 1954], Doc2Vec [Le and Mikolov, 2014], Latent Semantic Indexing (LSI) [LANDAUER and DUMAIS, 1997], Word2Vec [Mikolov et al., 2013a], in terms of classification accuracy, recall, precision, etc. Unlike traditional document representations which need very large corpus as input to create sparse representations and project them into a lower dimensional dense vector space, including Doc2Vec [Le and Mikolov, 2014] and Word2Vec [Mikolov et al., 2013b]. The proposed HWE model can use much less data with the help of existing knowledge resources like WordNet. Moreover, in the scenario where a document contains new words that have not appeared in the training set, both Word2Vec and Doc2Vec fail to capture the word semantics without re-training the entire corpus. If a document contains words that are not in the training set. Traditional models will either have to ignore the words or re-train the entire corpus. If the words can be found in WordNet, the proposed HWE model is more flexible to derive the semantics of new words as it can utilize WordNet to attribute new words to related concepts that have already been trained (AWE-O). More than often, new words will not appear in WordNet until they are verified and manually curated. In this case, HWE can also achieve better results by utilizing the containing text information (AWE-T) than Word2Vec or Doc2Vec. Nowadays, words are endowed with new semantic meanings rapidly in the light of large volumes of news feed, tweets, blogs, etc. This rapid semantic change or enrichment can not be well captured by a

single thesaurus or ontology. The proposed HWE can alleviate this problem by taking advantage of both context information as well as the pre-established structured knowledge in ontologies.

The benefits from enriching text content using ontologies such as WordNet is two-folded. First, it resolves synonyms. Words or phrases referring to the same concept are grouped together. Similar words like "fabulous", "fantastic" will be grouped as one automatically. Second, it produces more efficient and accurate text representations. The semantics of these words can be derived directly without knowledge of the contexts. These commonly already known concepts (context-free concepts) could be excluded in regular NLP systems. In lieu of training context-free concepts, we encoded these concepts with the aid of domain knowledge which could be considered the preprocessing of our proposed text representations. Not only this can reduce computational costs involved in the training process, but also it might better optimize the system.

The properties of the hybrid word embeddings could be very useful to perform NLP tasks. For example, other than representing document vectors in text classifications. It can also benefit document summarizations. Keywords can be extracted from word embeddings by finding most important embeddings for the documents which can be regarded as tags for the documents. One of the subjects where most NLP systems fail is short documents such as tweets, microblogs, etc. Limited information contained in these documents impedes the training of most NLP models. The domain knowledge incorporated into word embeddings can supplement the loss of information brought by the lack of context in short documents. Combining the domain knowledge with the context information encoded in the word embeddings are expected to alleviate the situation where synonmy and polysemy heavily reduce NLP system performances. Although the focus of this work is to represent documents, the

word embeddings generated from the proposed HWE model can be directly applied to tasks such as machine translation. With proper knowledge resources, such as abbreviation dictionaries, this work can also be applied to a wider range of scenarios, short message analysis, information retrieval, etc.

Appendices

Appendix A List of Acronyms and Abbreviations

AIC	-	Aggregate Information Content
API	-	Application Programming Interface
AWE-O	-	Accumulate Word Embeddings From Ontology
AWE-T	-	Accumulate Word Embeddings From Text
BOW	-	Bag Of Words
BP	-	Biological Process
CC	-	Cellular Component
CT	-	Concept Tree
DAG	-	Directed Acyclic Graph
DV	-	Document Vector
FN	-	False Negative
FP	-	False Positive
GO	-	Gene Ontology
HWE	-	Hybrid Word Embeddings

IC	-	Information Content
K-NN	-	K-Nearest Neighbors
LCA	-	Least Common Ancestor
LSA	-	Latent Semantic Analysis
LSI	-	Latent Semantic Indexing
MF	-	Molecular Function
MICA	-	Maximum Information Contained in Ancestors
MOD	-	Model Organism Database
NGD	-	Normalized Google Distance
NLP	-	Natural Language Processing
POS	-	Part Of Speech
PV-DBOW	-	Distributed Memory Model Paragraph Vectors
PV-DM	-	Distributed Bag Of Words Paragraph Vectors
SCR	-	Semantic Content Rate
SCV	-	Semantic Content Value
SV	-	Semantic Value

SVD	-	Singular Value Decomposition
SW	-	Semantic Weight
TF-IDF	-	Term Frequency - Inverted Document Frequency
TN	-	True Negative
TP	-	True Positive
TV	-	Tree Vector
VSM	-	Vector Space Model
WSCR	-	Weighted Semantic Content Rate
WSCV	-	Weighted Semantic Content Value
WSD	-	Word Sense Disambiguation

Appendix B Install and Run G-SESAME Website

B.1 Installation

1. Install Database.

   - Install MySQL or MariaDB.
     - The details follow the MySQL or MariaDB installation documents.
   - Get the latest GO databases at `archive.geneontology.org/full/latest/go_monthly-assocdb-tables.tar.gz`.
   - Use MySQL or MariaDB to import the data.

     tar -zxvf go-YYYYMM-TYPE-tables.gz

     cd ⟨releasedir⟩

     echo "create database mygo" | mysql

     cat *.sql | mysql mygo

     mysqlimport -L mygo *.txt

2. Install G-SESAME.

   - Put G-SESAME under /var/www/html/.
   - Change database connection credentials in database.php.

3. Install Apache.

   - The details follow the Apache installation documents.

4. Install PHP.

   - The details follow the PHP installation documents.

B.2 Startup the Web Application

1. Run Apache.

    - Stop Apache

      sudo systemctl stop httpd

    - Start Apache

      sudo systemctl start httpd

    - Start Apache at boot

      sudo systemctl enable httpd

    - Check the status of Apache

      sudo systemctl status httpd

2. Update statistics of GO terms. This will update GO terms statics that are used in the calculation of AIC.

    - cd /var/www/html/G-SESAME/Program/

      update.php

3. Open a web browser, enter `http://localhost/G-SESAME/index.php`.

Bibliography

[pyt, 2017] (2017). Pytorch, http://www.pytorch.org.

[Atkinson et al., 2009] Atkinson, J., Ferreira, A., and Aravena, E. (2009). Discovering implicit intention-level knowledge from natural-language texts. *Knowledge-Based Systems*, 22(7):502 – 508. Artificial Intelligence 2008.

[Azuaje et al., 2005] Azuaje, F., Wang, H., and Bodenreider, O. (2005). Ontology-driven similarity approaches to supporting gene functional assessment. *Proc. of the ISMB'2005 SIG meeting on Bio-ontologies*, pages 9–10.

[Banerjee and Pedersen, 2003] Banerjee, S. and Pedersen, T. (2003). Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, pages 805–810, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Bengio et al., 2003] Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155.

[Blei, 2012] Blei, D. M. (2012). Probabilistic topic models. *Commun. ACM*, 55(4):77–84.

[Bollegala et al., 2007] Bollegala, D., Matsuo, Y., and Ishizuka, M. (2007). Measuring semantic similarity between words using web search engines. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 757–766, New York, NY, USA. ACM.

[Budanitsky and Hirst, 2001] Budanitsky, A. and Hirst, G. (2001). Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *IN WORKSHOP ON WORDNET AND OTHER LEXICAL RESOURCES, SECOND MEETING OF THE NORTH AMERICAN CHAPTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*.

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

[Cheng et al., 2004] Cheng, J., Cline, M., Martin, J., Finkelstein, D., Awad, T., Kulp, D., and Siani-Rose, M. A. (2004). A knowledge-based clustering algorithm driven by gene ontology. *Journal of Biopharmaceutical Statistics*, 14(3):687–700.

[Chim and Deng, 2008] Chim, H. and Deng, X. (2008). Efficient phrase-based document similarity for clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(9):1217–1229.

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

[Cimiano, 2006] Cimiano, P. (2006). *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Collins and Quillian, 1969] Collins, A. M. and Quillian, M. R. (1969). Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*, 8(2):240 – 247.

[Collobert and Weston, 2008] Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA. ACM.

[Consortium, 2000] Consortium, T. G. O. (2000). Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29.

[Consortium, 2013] Consortium, T. G. O. (2013). Gene ontology annotations and resources. *Nucleic Acids Research*, 41(D1):D530–D535.

[Consortium, 2008] Consortium, T. U. (2008). The uniprot consortium: The universal protein resource (uniprot). *Nucleic Acids Research*, pages 190–195.

[Corley and Mihalcea, 2005] Corley, C. and Mihalcea, R. (2005). Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, EMSEE '05, pages 13–18, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Courrieu et al., 2010] Courrieu, P., Brand-D'abrescia, M., Peereman, R., and Rey, D. S. A. (2010). Validated intraclass correlation statistics to test item performance models. *Behavior Research Methods*.

[Curran, 2002] Curran, J. R. (2002). Ensemble methods for automatic thesaurus extraction. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 222–229, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Dhillon et al., 2003] Dhillon, I. S., Mallela, S., and Kumar, R. (2003). A divisive information theoretic feature clustering algorithm for text classification. *J. Mach. Learn. Res.*, 3:1265–1287.

[Ding et al., 2004] Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R. S., Peng, Y., Reddivari, P., Doshi, V., and Sachs, J. (2004). Swoogle: A search and metadata engine for the semantic web. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, CIKM '04, pages 652–659, New York, NY, USA. ACM.

[Du et al., 2009] Du, Z., Li, L., Chen, C.-F., Yu, P. S., and Wang, J. Z. (2009). G-sesame: web tools for go-term-based gene similarity analysis and knowledge discovery. *Nucleic Acids Research*, 37:W345–W349.

[Faria et al., 2007] Faria, D., Pesquita, C., Couto, F., and Falcao, A. (2007). A web tool for protein semantic similarity. *Department of Informatics, University of Lisbon.*

[Froehlich et al., 2007] Froehlich, H., Speer, N., Poustka, A., and Beissbarth, T. (2007). Gosim - an r-package for computation of information theoretic go similarities between terms and gene products. *BMC Bioinformatics*, 8:166.

[Gibbons and Roth, 2002] Gibbons, F. D. and Roth, F. P. (2002). Judging the quality of gene expression-based clustering methods using gene annotation. *Genome Research*, 12:1574–1581.

[Golub and Reinsch, 1970] Golub, G. H. and Reinsch, C. (1970). Singular value decomposition and least squares solutions. *Numer. Math.*, 14(5):403–420.

[Hadj Taieb et al., 2013] Hadj Taieb, M. A., Ben Aouicha, M., and Ben Hamadou, A. (2013). Computing semantic relatedness using wikipedia features. *Know.-Based Syst.*, 50:260–278.

[HadjTaieb et al., 2014] HadjTaieb, M. A., Ben Aouicha, M., and Ben Hamadou, A. (2014). A new semantic relatedness measurement using wordnet features. *Knowledge and Information Systems*, 41(2):467–497.

[Harris and Harris, 2012] Harris, D. and Harris, S. (2012). *Digital Design and Computer Architecture, Second Edition.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition.

[Harris, 1954] Harris, Z. (1954). Distributional structure. *Word*, 10(23):146–162.

[Heyer et al., 1999] Heyer, L. J., Kruglyak, S., and Yooseph, S. (1999). Exploring expression data: Identification and analysis of coexpressed genes. *Genome Research*, 9:1106–1115.

[Horrocks, 2008] Horrocks, I. (2008). Ontologies and the semantic web. *Commun. ACM*, 51(12):58–67.

[Hotho et al., 2003] Hotho, A., Staab, S., and Stumme, G. (2003). Wordnet improves text document clustering. In *In Proc. of the SIGIR 2003 Semantic Web Workshop*, pages 541–544.

[Jiang et al., 2004] Jiang, D., Tang, C., and Zhang, A. (2004). Cluster analysis for gene expression data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 16:1370–1386.

[Jiang and Conrath, 1997] Jiang, J. J. and Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. *Proc. Int. Conf. on Research in Computational Linguistics*, pages 19–33.

[Kobayashi and Takeda, 2000] Kobayashi, M. and Takeda, K. (2000). Information retrieval on the web. *ACM Comput. Surv.*, 32(2):144–173.

[Kogan et al., 2005] Kogan, J., Teboulle, M., and Nicholas, C. (2005). Data driven similarity measures for k-means like clustering algorithms. *Inf. Retr.*, 8(2):331–349.

[Kriventseva et al., 2001] Kriventseva, E. V., Fleischmann, W., Zdobnov, E. M., and Apweiler, R. (2001). Clustr: a database of clusters of swiss-prot+trembl proteins. *Nucleic Acids Research*, 29:33–36.

[LANDAUER and DUMAIS, 1997] LANDAUER, T. and DUMAIS, S. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211–240.

[Landauer et al., 1998] Landauer, T., Foltz, P., and Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, 25:259–284.

[Le and Mikolov, 2014] Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In Jebara, T. and Xing, E. P., editors, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196. JMLR Workshop and Conference Proceedings.

[Leacock and Chodorow, 1998] Leacock, C. and Chodorow, M. (1998). Combining local context and wordnet similarity for word sense identification. In *WordNet: An Electronic Lexical Database*, volume 49, pages 265–.

[Leacock et al., 1998] Leacock, C., Miller, G. A., and Chodorow, M. (1998). Using corpus statistics and wordnet relations for sense identification. *Comput. Linguist.*, 24(1):147–165.

[Lesk, 1986] Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, pages 24–26, New York, NY, USA. ACM.

[Li et al., 2010] Li, B., Wang, J. Z., Feltus, F. A., Zhou, J., and Luo, F. (2010). Effectively integrating information content and structural relationship to improve the go-based similarity measure between proteins. *CoRR*, abs/1001.0958.

[Li et al., 2013] Li, M., Wu, X., Pan, Y., and Wang, J. (2013). hf-measure: A new measurement for evaluating clusters in proteinprotein interaction networks. *PROTEOMICS*, 13(2):291–300.

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

[Li and Lu, 2008] Li, Y. and Lu, B.-L. (2008). Semantic similarity definition over gene ontology by further mining of the information content. *APBC*, 6:155–164.

[Lin, 1998] Lin, D. (1998). An information-theoretic definition of similarity. In *In Proceedings of the 15th International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann.

[Lin et al., 2014] Lin, Y. S., Jiang, J. Y., and Lee, S. J. (2014). A similarity measure for text classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 26(7):1575–1590.

[Lodhi et al., 2002] Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. (2002). Text classification using string kernels. *J. Mach. Learn. Res.*, 2:419–444.

[Loper and Bird, 2002] Loper, E. and Bird, S. (2002). Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

[Martin et al., 2004] Martin, D., Brun, C., Remy, E., Mouren, P., Thieffry, D., and Jacq, B. (2004). GOToolBox: functional analysis of gene datasets based on Gene Ontology. *Genome Biology*, 5(12):R101+.

[Mikolov et al., 2013a] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

[Mikolov et al., 2013b] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

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

[Olivera,] Olivera, F. Memory systems in organizations: An empirical investigation of mechanisms for knowledge collection, storage and access. *Journal of Management Studies*, 37(6):811–832.

[Ovaska et al., 2008] Ovaska, K., Laakso, M., and Hautaniemi, S. (2008). Fast Gene Ontology based clustering for microarray experiments. *BioData Mining*, 1(1):11+.

[Patwardhan, 2006] Patwardhan, S. (2006). Using wordnet-based context vectors to estimate the semantic relatedness of concepts. In *In: Proceedings of the EACL*, pages 1–8.

[Patwardhan, 2003] Patwardhan, S. V. (2003). Incorporating dictionary and corpus information into a context vector measure of semantic relatedness.

[Pekar and Staab, 2002] Pekar, V. and Staab, S. (2002). Taxonomy learning: factoring the structure of a taxonomy into a semantic classification decision. *Proc. Int. Conf. on Computational Linguistics*, 2:786–792.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

[Pesquita et al., 2007] Pesquita, C., Faria, D., Bastos, H., Falcao, A. O., and Couto, F. M. (2007). Evaluating go-based semantic similarity measures. *Proc. of the 10th Annual Bio-Ontologies Meeting 2007*, pages 37–40.

[Pesquita et al., 2009] Pesquita, C., Faria, D., Falcao, A. O., Lord, P., and Couto, F. M. (2009). Semantic similarity in biomedical ontologies. *PLoS Computational Biology*, 5(7):e1000443.

[Petrakis et al., 2006] Petrakis, E. G. M., Varelas, G., Hliaoutakis, A., and Raftopoulou, P. (2006). X-similarity: Computing semantic similarity between concepts from different ontologies. *Journal of Digital Information Management (JDIM*, 4.

[Quillian, 1969] Quillian, M. R. (1969). The teachable language comprehender: A simulation program and theory of language. *Commun. ACM*, 12(8):459–476.

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

[Ravasi et al., 2010] Ravasi, T., Suzuki, H., Vittorio Cannistraci, C., Katayama, S., Bajic, V., Tan, K., Akalin, A., Schmeier, S., Kanamori-Katayama, M., Bertin, N., Carninci, P., Daub, C., Forrest, A., Gough, J., Grimmond, S., Han, J.-H., Hashimoto, T., Hide, W., Hofmann, O., and Hayashizaki, Y. (2010). An atlas of combinatorial transcriptional regulation in mouse and man. 141:369–369.

[Řehůřek and Sojka, 2010] Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. `http://is.muni.cz/publication/884893/en`.

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

[Rocchio, 1971] Rocchio, J. J. (1971). Relevance feedback in information retrieval. In Salton, G., editor, *The Smart retrieval system - experiments in automatic document processing*, pages 313–323. Englewood Cliffs, NJ: Prentice-Hall.

[Rodríguez and Egenhofer, 2003] Rodríguez, M. A. and Egenhofer, M. J. (2003). Determining semantic similarity among entity classes from different ontologies. *IEEE Trans. on Knowl. and Data Eng.*, 15(2):442–456.

[Rong, 2014] Rong, X. (2014). word2vec parameter learning explained. *CoRR*, abs/1411.2738.

[Salton and Buckley, 1988] Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513 – 523.

[Salton and Lesk, 1968] Salton, G. and Lesk, M. E. (1968). Computer evaluation of indexing and text processing. *J. ACM*, 15(1):8–36.

[Salton and McGill, 1986] Salton, G. and McGill, M. J. (1986). Introduction to modern information retrieval.

[Salton et al., 1997] Salton, G., Singhal, A., Mitra, M., and Buckley, C. (1997). Automatic text structuring and summarization. *Information Processing & Management*, 33(2):193 – 207. Methods and Tools for the Automatic Construction of Hypertext.

[Schlicker and Albrecht, 2008] Schlicker, A. and Albrecht, M. (2008). Funsimmat: a comprehensive functional similarity database. *Nucleic Acids Research*, 36(Database-Issue):434–439.

[Schlicker et al., 2006] Schlicker, A., Domingues, F. S., Rahnenfuhrer, J., and Lengauer, T. (2006). A new measure for functional similarity functional similarity of gene products based on gene ontology. *BMC Bioinformatics*, 7:302.

[Sebti and Barfroush, 2008] Sebti, A. and Barfroush, A. A. (2008). A new word sense similarity measure in wordnet. In *2008 International Multiconference on Computer Science and Information Technology*, pages 369–373.

[Seco et al., 2004] Seco, N., Veale, T., and Hayes, J. (2004). An intrinsic information content metric for semantic similarity in wordnet. In *Proceedings of the 16th European Conference on Artificial Intelligence*, ECAI'04, pages 1089–1090, Amsterdam, The Netherlands, The Netherlands. IOS Press.

[Sevilla et al., 2005] Sevilla, J. L., Segura, V., Podhorski, A., Guruceaga, E., Mato, J. M., Martinez-Cruz, L. A., Corrales, F. J., and Rubio, A. (2005). Correlation between gene expression and go semantic similarity. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2:330–338.

[Simov et al., 2016] Simov, K., Osenova, P., and Popov, A. (2016). Using context information for knowledge-based word sense disambiguation. In Dichev, C. and Agre, G., editors, *Artificial Intelligence: Methodology, Systems, and Applications*, pages 130–139, Cham. Springer International Publishing.

[Smialowski et al., 2010] Smialowski, P., Pagel, P., Wong, P., Brauner, B., Dunger, I., Fobo, G., Frishman, G., Montrone, C., Rattei, T., Frishman, D., and Ruepp, A. (2010). The negatome database: a reference set of non-interacting protein pairs. *Nucleic Acids Research*, 38:D540–D544.

[Song et al., 2013] Song, X., Li, L., Srimani, P. K., Yu, P. S., and Wang, J. Z. (2013). Measure the semantic similarity of go terms using aggregate information content. In Cai, Z., Eulenstein, O., Janies, D., and Schwartz, D., editors, *Bioinformatics Research and Applications*, volume 7875 of *Lecture Notes in Computer Science*, pages 224–236. Springer Berlin Heidelberg.

[Song et al., 2014] Song, X., Li, L., Srimani, P. K., Yu, P. S., and Wang, J. Z. (2014). Measure the semantic similarity of go terms using aggregate information content. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 11(3):468–476.

[Spellman et al., 1998] Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D., and Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, 9:3273–3297.

[Stein et al., 2002] Stein, L. D., Mungall, C., Shu, S., Caudy, M., Mangone, M., Day, A., Nickerson, E., Stajich, J. E., Harris, T. W., Arva, A., and Lewis, S. (2002). The

generic genome browser: A building block for a model organism system database. *Genome Research*, 12:1599–1610.

[Stevenson and Greenwood, 2005] Stevenson, M. and Greenwood, M. A. (2005). A semantic approach to ie pattern induction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 379–386, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Sun and Lim, 2001] Sun, A. and Lim, E.-P. (2001). Hierarchical text classification and evaluation. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 521–528.

[Snchez and Batet, 2011] Snchez, D. and Batet, M. (2011). Semantic similarity estimation in the biomedical domain: An ontology-based information-theoretic perspective. *Journal of Biomedical Informatics*, 44(5):749 – 759.

[Snchez et al., 2012] Snchez, D., Batet, M., Isern, D., and Valls, A. (2012). Ontology-based semantic similarity: A new feature-based approach. *Expert Systems with Applications*, 39(9):7718 – 7728.

[Taieb et al., 2014] Taieb, M. A. H., Aouicha, M. B., and Hamadou, A. B. (2014). Ontology-based approach for measuring semantic similarity. *Engineering Applications of Artificial Intelligence*, 36:238 – 261.

[Teng et al., 2013] Teng, Z., Guo, M., Liu, X., Dai, Q., Wang, C., and Xuan, P. (2013). Measuring gene functional similarity based on group-wise comparison of go terms. *Bioinformatics*, 29(11):1424–1432.

[Varelas et al., 2005] Varelas, G., Voutsakis, E., Raftopoulou, P., Petrakis, E. G., and Milios, E. E. (2005). Semantic similarity methods in wordnet and their application to information retrieval on the web. In *Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management*, WIDM '05, pages 10–16, New York, NY, USA. ACM.

[Voorhees, 1993] Voorhees, E. M. (1993). Using wordnet to disambiguate word senses for text retrieval. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '93, pages 171–180, New York, NY, USA. ACM.

[Wang et al., 2004] Wang, H., Azuaje, F., Bodenreider, O., and Dopazo, J. (2004). Gene expression correlation and gene ontology-based similarity: An assessment of quantitative relationships. *Proc. of the 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 25–31.

[Wang et al., 2007] Wang, J. Z., Du, Z., Payattakool, R., Yu, P. S., and Chen, C.-F. (2007). A new method to measure the semantic similarity of go terms. *Bioinformatics*, 23(10):1274–1281.

[Wang and Taylor, 2007] Wang, J. Z. and Taylor, W. (2007). Concept forest: A new ontology-assisted text document similarity measurement method. In *Web Intelligence, IEEE/WIC/ACM International Conference on*, pages 395–401.

[Washington et al., 2009] Washington, N. L., Haendel, M. A., Mungall, C. J., Ashburner, M., Westerfield, M., and Lewis, S. E. (2009). Linking human diseases to animal models using ontology-based phenotype annotation. *PLoS biology*, 7(11):e1000247.

[Wu et al., 2005] Wu, H., Su, Z., Mao, F., Olman, V., and Xu, Y. (2005). Prediction of functional modules based on comparative genome analysis and gene ontology application. *Nucleic Acids Research*, 33(9):2822–2837.

[Wu and Palmer, 1994] Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Xu and Croft, 1996] Xu, J. and Croft, W. B. (1996). Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96, pages 4–11, New York, NY, USA. ACM.

[Xu et al., 2008] Xu, T., Du, L., and Zhou, Y. (2008). Evaluation of go-based functional similarity measures using s.cerevisiae protein interaction and expression profile data. *BMC Bioinformatics*, 9:472.

[Yang et al., 2012] Yang, H., Nepusz, T., and Paccanaro, A. (2012). Improving GO semantic similarity measures by exploring the ontology beneath the terms and modelling uncertainty. *Bioinformatics*, 28(10):1383–1389.

[Yu et al., 2010] Yu, G., Li, F., Qin, Y., Bo, X., Wu, Y., and Wang, S. (2010). Gosemsim: an r package for measuring semantic similarity among go terms and gene products. *Bioinformatics*, 26(7):976–978.

[Yuan et al., 2013] Yuan, Q., Yu, Z., and Wang, K. (2013). A new model of information content for measuring the semantic similarity between concepts. In *2013 International Conference on Cloud Computing and Big Data*, pages 141–146.

[Zhou et al., 2008a] Zhou, Z., Wang, Y., and Gu, J. (2008a). A new model of information content for semantic similarity in wordnet. In *2008 Second International*

Conference on Future Generation Communication and Networking Symposia, volume 3, pages 85–89.

[Zhou et al., 2008b] Zhou, Z., Wang, Y., and Gu, J. (2008b). New model of semantic similarity measuring in wordnet. In *2008 3rd International Conference on Intelligent System and Knowledge Engineering*, volume 1, pages 256–261.