Acknowledgments

First and foremost, I offer my sincerest gratitude to my supervisor, Dr. Srikanta Bedathur, who has supported me throughout my thesis with his patience and knowledge whilst allowing me the room to work in my own way.

I dedicate this thesis to my family who have always supported me and believed that I could do it.

I would like to thank all my friends for their invaluable help and moral support.

Keywords: Wikipedia, Semantic Similarity, Hierarchical Abstraction

Certificate

This is to certify that the thesis titled **Semantic Similarity through Hierarchical Abstraction of Knowledge** submitted by **Kanchan Arora** for the partial fulfilment of the requirements for the degree of *Master of Technology* in *Computer Science & Engineering* is a record of the bonafide work carried out by her under my guidance and supervision in the Information Management and Data Analytics group at Indraprastha Institute of Information Technology, Delhi. This work has not been submitted anywhere else for the reward of any other degree.

Dr. Srikanta Bedathur
Assistant Professor
Indraprastha Institute of Information Technology, New Delhi

Abstract

Identifying semantic similarity between two texts has many applications in NLP including information extraction and retrieval, word sense disambiguation, text summarization and type classification. Similarity between texts is commonly determined using a taxonomy based approach, but the limited scalability of existing taxonomies has led recent research to use Wikipedia's encyclopaedic knowledge base to find similarity or relatedness. In this thesis, we propose Hierarchical Semantic Analysis, a method which represents semantics of a text in high dimensional space of Wikipedia concepts and category hierarchies. We represent the meaning of any text excerpt as a weighed vector of Wikipedia-based resources. To evaluate the similarity of texts in this space, we compare the corresponding vectors using conventional metrics (e.g. cosine). Compared with the previous state of the art, use of Hierarchical Semantic Analysis(HSA) results in substantial improvements in correlation of computed similarity scores with human judgements from r= .873 to 0.901 for short sentence pairs and from r= .72 to 0.863 for paragraph pairs.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Semantic Similarity

Similarity refers to psychological nearness between two concepts[4]. Similarity has roots in psychology, social sciences, mathematics, physics and computer science.Semantic similarity or semantic relatedness is the idea of estimating distance between two texts which is based on the sameness of their meaning or semantic content rather than on their syntactical resemblance. How much two concepts *"cemetery"* and *'graveyard'* are similar? Is the statement *"Nigerian President Olusegun Obasanjo said he will weep if a single mother sentenced to death by stoning for having a child out of wedlock is killed"* and statement *"An Islamic high court in northern Nigeria rejected an appeal today by a single mother sentenced to be stoned to death for having sex out of wedlock"* convey similar message?. Judgements for semantic likeness between natural language comments is commonly performed by humans but remains a challenge for machines as humans interpret the words of a document in the much larger context of their prior knowledge. It has long been recognized that in order to process natural language, computers require prudence and access to vast amounts of domain-specific knowledge.

The classification of semantic similarity measures include similarity measures for single ontology and multiple ontologies. The classification is based on how the semantic similarity measure is quantified. The quantification is either based on the ontological structure or based on the information content.

1.1.1 Semantic similarity based on single ontology

Similarity between concepts belonging to single ontology have different approaches such as

-Path based measure

-Information content based measure
-Hybrid measure
-Feature based measure

Based on the quantifying similarity approaches are used for the semantic measure. Also in some cases both path length based and information content based approaches have been used.

Path based measure

The similarity measurement between concepts is based on the path distance separating the concepts. In this measure the quantification of similarity is based on the ontology or taxonomy structure. In these ontology or taxonomical structure, most predominant relations are connected through is-a type relation. Thus similarity is computed by shortest path and the degree of similarity is determined based on path length. The various path based similarity measures are

Wu and Palmer measure: Wu and Palmer introduced a scaled measure [22].This similarity measure takes the position of concepts c1 and c2 in the taxonomy relatively to the position of the most specific common concept lcs(c1,c2) into account. It assumes that the similarity between two concepts is the function of path length and depth in path-based measures.

$$sim_{WP}(c_1, c_2) = \frac{2 * depth(lcs(c_1, c_2))}{len(c_1, c_2) + 2 * depth(lcs(c_1, c_2))} \tag{1.1}$$

where len(c_i,c_j): the length of the shortest path from synset c_i to synset c_j in WordNet.
lcs(c_i,c_j): the lowest common subsumer of c_i and c_j
depth(c_i): the length of the path to synset c_i from the global root entity, and depth(root)=1.
From formula it is noted that,
(1)The similarity between two concepts (c_1, c_2) is the function of their distance and the lowest common subsumer(lcs(c_1,c_2)).
(2)If the lcs(c_1,c_2) is root,depth(lcs(c_1,c_2))=1, $sim_{\mathrm{WP}}(c_1,c_2)$ >0;
if the two concepts have the same sense, the concept c_1, concept c_2 and lcs(c_1,c_2) are the same node len(c_1,c_2)=0,$sim_{\mathrm{WP}}(c_1,c_2)$ = 1 ;
otherwise 0<$depth(lso(c_1, c_2))$<$deep\_max$
where deep_max is the the max depth(c_i) of the taxonomy 0<$len(c_1, c_2)$<2 * $deep\_max$, 0<$sim_{WP}(c_1, c_2)$<1. Thus, the values of $sim_{\mathrm{WP}}(c_1,c_2)$ are in(0,1].

Leacock and Chodorow Similarity measure:
Leakcock and Chodorow took the maximum depth of taxonomy into account

and proposed the following measure [8]:

$$sim_{LC}(c_1, c_2) = -\log \frac{len(c_1, c_2)}{2 * deep\_max} \tag{1.2}$$

From formula it is noted that,
(1) As same as formula (1) the similarity between two concepts (c_1, c_2) is the function of the shortest path len(c_1,c_2) from c_1 to c_2.
(2) When c_1 and c_2 have the same sense, len(c_1,c_2) =0. In practice, we add 1 to both len(c_1,c_2) and 2*deep_max to avoid log (0). Thus the values of sim_{LC}(c_1,c_2) are in (0, log(2*deep_max+1)]

Information content based measure

Both the path length and depth relative measure use the knowledge solely captured by ontology to computationally determine the similarity between concepts. In this section the knowledge revealed by corpus is used to augment the information already present in the ontologies or taxonomy. Thus information content based approach is also referred as the corpus based approach or information theoretic based approach. The various information content based measures are

Resnik Measure:In 1995 Resnik proposed information content-based similarity measure [15]. It assumes that for two given concepts, similarity is depended on the information content that subsumes them in the taxonomy.

$$sim_{Resnik} = \log p(lcs(c_1, c_2) = IC(lcs(c_1, c_2)) \tag{1.3}$$

where p(c) is the probability of encountering an instance of concept c in large corpus. From formula it is noted that,the values only rely on concept pairs lowest subsumer in the taxonomy.Probability of a concept was estimated as:

$$p(c) = \frac{freq(c)}{N} \tag{1.4}$$

where N is the total number of nouns, and freq(c) is the frequency of instance of concept c occurring in the taxonomy. When computing freq(c), each noun or any of its taxonomical hyponyms that occurred in the given corpora is included, which implies that if c1 is-a c2, then p(c1) <p(c2). Thus the more abstract the concept is, the higher its associated probability and the lower its information content.

$$freq(c) = \sum_{w \in W(c)} count(w) \tag{1.5}$$

Lin Measure:Lin proposed another method for similarity measure [12] It uses both the amount of information needed to state the commonality

between the two concepts and the information needed to fully describe these terms.

$$sim_{Lin}(c_1, c_2) = \frac{2 * IC(lcs(c_1, c_2))}{IC(c_1) + IC(c_2)} \tag{1.6}$$

From formula it is noted that,the measure has taken the information content of compared concepts into account respectively. As $IC(lcs(c_1, c_2)) \leq IC(c_1)$ and $IC(lcs(c_1, c_2)) \leq IC(c_2)$, therefore the values of this measure vary between 1 and 0.

Jiang and Conrath measure: Jiang calculated semantic distance to obtain semantic similarity [6]. Semantic similarity is the opposite of the distance.

$$dis_{Jaing}(c_1, c_2) = IC(c_1) + IC(c_2) - 2IC(lcs(c_1, c_2)) \tag{1.7}$$

From formula it is noted that, (1) The measure has taken the IC of compared concepts into account respectively.
(2) The value is semantic distance between two concepts. Semantic similarity is the opposite of the semantic distance.

Hybrid measure

Hybrid combines knowledge derived from various sources of information. The major advantage of these approaches is if the knowledge of an information source is inadequate then it may be derived from the alternate information sources.The various hybrid similarity measures are

Li Measure: Li's measure is intuitively and empirically derived [10]. It is based on the assumption that information sources are infinite to some extend while humans compare word similarity with a finite interval between completely similar and nothing similar. Intuitively the transformation between an infinite interval to a finite one is nonlinear. Therefore the measure combines the shortest path and the depth of concepts in a non-linear function:

$$sim_{Li}(c1, c2) = e^{-\alpha * len(c1,c2)} \frac{e^{\beta * depth(lcs(c1,c2))} - e^{-\beta * depth(lcs(c1,c2))}}{e^{\beta * depth(lcs(c1,c2))} + e^{-\beta * depth(lcs(c1,c2))}} \tag{1.8}$$

From formula it is noted that,
(1) Formula will monotonically increasing with respect to depth(lso(c_1,c_2)) and decreasing with len(c_1,c_2).
(2) If len(c_1,c_2) =0 and $depth(lso(c_1, c_2)) \longrightarrow deep\_max$, $sim_{Li}(c_1, c_2) \longrightarrow 1$.
if $len((c1, c2) \longrightarrow 2 * deep\_max$ and depth(lso(c1,c2))=1,
$sim_{Li}(c1, c2) \longrightarrow 0$. Thus the values of sim_{Li} (c_1,c_2) are in (0, 1) .
(3) Parameter and need to be adapted manually for good performance.

Zhou measure: Zhou has proposed a measure[23]

$$sim\_Zhou(c_1, c_2) = 1 - k(\frac{\log len(c_1, c_2) + 1}{\log 2 * deep\_max - 1}) - (1 - k) * ((IC(c_1) + IC(c_2) - 2 * IC(lcs(c_1, c_2))/2) \quad (1.9)$$

From formula it is noted that,
(1) both IC and path have been taken into considerate.
(2) parameter k needs to be adapted manually for good performance. If k=1, formula is path-based; if k=0, formula is IC-based measure.

Feature based Measure

Feature based approach takes into account the features that are familiar to both concepts and also the specific differentiating features of each concept. Thus the various feature based measures are

Tversky measure: One classical measure is Tversky's model, which argues that similarity is not symmetric. Features between a subclass and its superclass have a larger contribution to the similarity evaluation than those in the inverse direction. It is defined as [20]:

$$sim_{Tversky}(c_1, c_2) = \frac{|C_1 \cap C_2|}{|C_1 \cap C_2| + k|C_1/C_2| + (k-1)|C_2/C_1|} \quad (1.10)$$

Where C_1, C_2 correspond to description sets of concept c_1 and $c2$ respectively, k is adjustable and $k \in [0, 1]$. From formula (19) it is noted that, (1) the values of $sim_{\text{Tversky}}(c_1,c_2)$ vary from 0 to 1. (2) $sim_{\text{Tversky}}(c_1,c_2)$ increases with commonality and decreases with the difference between the two concepts.

1.2 Semantic similarity based on multiple ontology

The semantic similarity measures discussed earlier are meant single ontology. Now in recent days with the growing information sources on the web, there is a need for developing measures which will compute similarity among concepts belonging to different ontologies.Similarity measures between concepts in multiple ontology is classified as

-Path length based measure
-Feature based measure

Cross ontology measures compare the words from different ontology. The cross ontology often requires hybrid or feature based measure, because the

structure and information content between diverse ontologies cannot be compared directly.Cross ontology measure includes the following steps:

-Extracting set of relevant definitions, features, synsets and neighbors from both ontology

-Word matching
-Feature matching
-Semantic neighbourhood matching
-Finding cross ontology measure for the input query

1.3 Methods of Determining Semantic Similarity

1.3.1 Wikipedia as a Taxonomy

Wikipedia contains over 4.4 million articles in the English version alone as of March,2014. It has recently provided a wide range of knowledge including some proper nouns in different areas of expertise which is not described in WordNet .It also includes a large number of articles about almost every entity in the world. Wikipedia provides a semantic network for semantic relatedness in a more structured fashion than a search engine with more coverage than WordNet. This feature provides the hierarchical structure or network. Wikipedia also provides articles link graph.

Being collaborative and open effort always means that Wikipedia has up to date information, but this means that structure is not as well defined as a controlled taxonomy. This ill-defined structure introduces variability that needs to be accounted for. The variability could be as simple as a mistake in classifying an article into a category, or may be as severe as deliberate vandalism, and in either case can affect the similarity or relatedness measure.

1.3.2 Mapping of Concepts

In all of the methods examined in this thesis document, Wikipedia is used as the taxonomy for determining the semantic similarity or semantic relatedness measurement. Since in NLP applications, the measurement is between two concepts pairs, and Wikipedia consists of articles, the concepts must be mapped to Wikipedia. This usually means that the concept is equated with a single article from Wikipedia that best matches the context. Some methods map the concepts automatically and some use a manual mapping. Regardless of whether it is done manually or automatically, there are three types of mapping that take place.

First is a single direct correspondence, where a concept directly corresponds to an article in Wikipedia. An example of this mapping would be to map the concept *"Car"* to Wikipedia. Wikipedia does not have an article named *"Car"*, but it does have an *"Automobile"* article that the concept directly corresponds to.

Second, a concept may apply to more than one article in Wikipedia, and may have to be disambiguated based on the context. For example, the surface forms used in a text may refer to more than one concept.For instance,surface form *"Srinath"* could refer to the Wikipedia article of *Javagal Srinath* and also *Srinath(Kannada Actor)*.

Finally, a concept may not apply directly to any article in Wikipedia, and has to be mapped to an article in Wikipedia that is similar or encapsulates the concept. This would be the case for concept *"String"*, which does not have an article in Wikipedia. The concept is encapsulated by the article *"Rope"*.

While time consuming, the human judgements involved in manually mapping concepts to Wikipedia articles is typically very accurate. Automatic mapping of the concept to the article is much more prone to mistakes.

The methods described in this thesis do mapping of concept to Wikipedia article using DBpedia Spotlight.

1.4 Wikipedia Resource

A text can be represented as conjunction of named entities mentioned in the text. Each article in a Wikipedia usually describes a named entity. Named entities are sequences of words that identify an entity. Example of named entities are Rahul Dravid , IIIT, USA. Each named entity falls into a particular type in a named entity hierarchy . These types are broad categories like person, location or organisation.

1.5 Overview of our method:HSA

A method of determining semantic similarity measures between two texts: **HSA(Hierarchical Semantic Analysis)**. HSA works in three major stages: first it recognizes Wikipedia resources in two texts ad then fetches category hierarchy for each identified resource and creates a weighed vector from the category hierarchies for both the texts and then finally performs a configurable vector comparison between vectors, yielding a similarity measurement score for the given texts. The implemented vector comparisons are well known in information retrieval: Cosine Similarity

1.5.1 Wikipedia Category Hierarchy

The Wikipedia category system forms a type of taxonomy where the collaborative tagging articles into different categories both categorizes and provides means to connect articles together. Any given category is allowed to have one or more subcategories or supercategories which forms a hierarchy of categories.

1.5.2 Intuition behind HSA

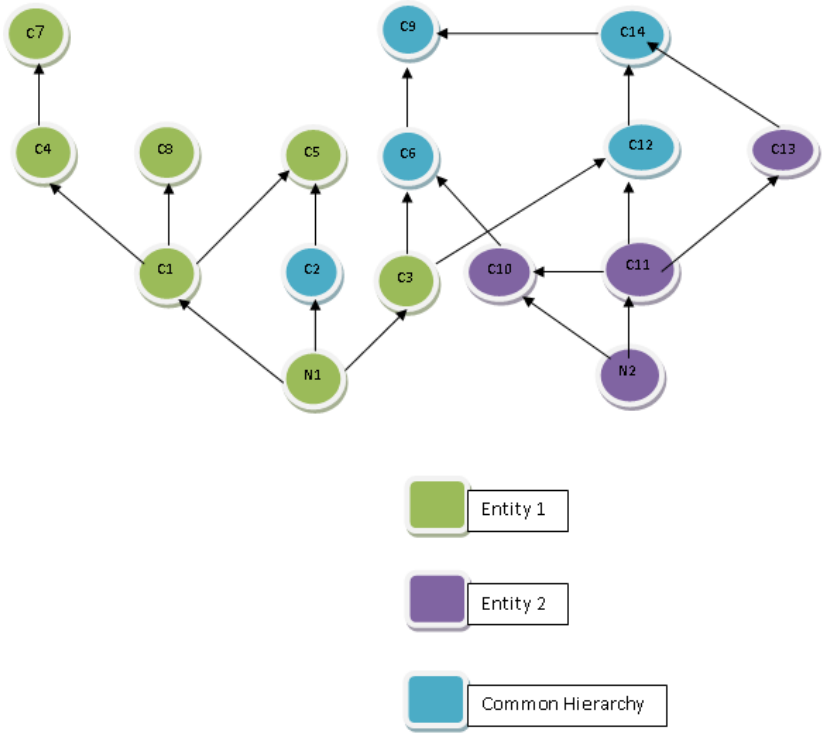
When two articles have categories in common, even super-categories up several levels, it follows that the articles are similar, as they have been tagged to belong to the same category hierarchies. This forms the basis on which the different comparison algorithms implemented for HSA operate. In HSA, the category hierarchies of two texts are compared together to determine a similarity score (Figure 1.1). The more categories that two category hierarchies have in common, the greater the similarity.

To determine the similarity measure, HSA first constructs a category vector for each of the texts containing all of the categories that are in that texts category hierarchy, and then performs a standard vector comparison method on the two resultant category vectors. The vector comparison method well known in information retrieval and data mining are Dice's Coefficient, Jaccard's Coefficient, and the cosine angle similarity measurement. We have calculated similarity score using all three comparison methods.

First, category vectors must be created for both N1 and N2. For N1, starting at depth 1, we have C1, C2 and C3. Depth 2 has C4, C8, C5, C6, and C12. Depth 3 consists of C7, C9 and C14, while the hierarchy ends at depth 4 with C9. Moving through the depths and assigning weight functions to those categories, N1s category vector is [C1:1, C2:1, C3:1, C4:0.73, C8:0.73, C5:0.73, C6:0.73, C12:0.73,C7:0.46, C9:0.46, C14:0.46]. C9 at depth of 4 is not included in the category vector as that category already exists for the hierarchy at a depth of 3. In the same way, the category vector for N2 is[C2:1, C10:1, C11:1, C6:0.73, C12:0.73, C13:0.73, C14:0.46, C9:0.46].

Once we have our category vectors, apply Cosine Similarity to the category vectors. For cosine measure, we need the product of the category vectors divided by the product of the mode of the category vectors to get the similarity value. The product of the category vectors is the set of all common categories, with an product of their weighted value. The product set for our example is [C2:1, C6:0.5329, C12:0.5329, C9:0.2116, C14:0.2116]. If we sum the values for the product set we get a value of 2.489. To calculate the mode of the category vectors, add square of weights of each of the categories from both articles category vectors. In our example the product of mode of

Figure 1.1: Comparing Category Hierarchies



two vectors is : 7.176 Following the Cosine based algorithm and formula, we divide the product set value by the mode value, 2.489/7.176, or 0.346, which is the similarity score using the Cosine Similarity for the example hierarchies in Figure 1.1.

Chapter 2

Related Work

2.1 Non-Wikipedia Based Semantic Similarity Measures

The characteristic flexibility of natural language that enables humans to express similar meanings using quite different sentences in terms of structure and length means that two sentences may be semantically related while containing no words in common. Consequently, a number of sentence similarity measures have been proposed.

Islam and Inkpen[5] proposed Semantic Text Similarity(STS) method which determines the similarity of two texts between semantic and syntactic information that they contain. Three similarity functions are considered : first, string similarity and semantic word similarity are calculated and then an optimal common-word order similarity function is used to incorporate syntactic information . Finally, the text similarity is derived by combining string similarity, semantic similarity and common-word order similarity with normalization.

Li[11] presents an algorithm that takes account of semantic information and word order information implied in the sentences. The semantic similarity of two sentences is calculated using information from a structured lexical database and from corpus statistics.

Feng[2]estimated the sentence similarity with consideration of direct relevance and indirect relevance between sentence pairs. The direct relevance is the means by which a human can get the obvious coherence between two concepts, whereas the indirect relevance is the means by which a human can get some potential relatedness between two concepts.

Kennedy[7] proposed a method of sentence representation that at-

tempts to leverage the structure found in Rogets Thesaurus and similar lexical ontologies and determine semantic relatedness between pairs of terms.

2.2 Wikipedia Based Semantic Similarity Measures

There are several approaches for measuring semantic relatedness using resources such as WordNet or Wikipedia categories as a graph or network by considering the number or length of paths between concepts. In the WikiRelate project, Ponzetto and Strube used three measures for computing semantic relatedness: First, a path-based measure using the length of the path between two concepts; second, an information content-based measure and third, the overlap-based measure which applies the Lesk algorithm that defines the relatedness between two words as a function of the overlap between two contexts defining the corresponding words. In WikiRelate[19], a pair of Wikipedia pages is first retrieved, then categories they refer to are extracted and finally, the relatedness between two concepts is computed regarding the paths found between two concepts in theWikipedia categories. In the last step, Ponzetto and Strube calculate relatedness by selecting the shortest path and the paths which maximize the information content-based measure.

In contrast with statistical methods for computing relatedness such as LSA, Gabrilovich and Markovitch proposed Explicit Semantic Analysis (ESA) using meaning in natural concepts derived from Wikipedia [3]. In this work, they used Wikipedia articles for augmenting the text representation and constructing a weighted list of concepts. They finally used tf-idf and conventional machine learning methods to calculate relatedness between the weighted vectors constructed in the previous steps.

Milne and Witten used cross referencing in the Wikipedia link database in order to obtain semantic relatedness between two concepts called Wikipedia Link-based Measure (WLM)[21]. In WLM, they used tf-idf using link counts weighted by the probability of occurrence of a term in an article. Almost all of the previous research has used tf-idf and VSM to calculate the relatedness between two sets of features.

Ruiz-Casado[17] proposed a procedure for automating the extension of an existing ontology or lexical semantic network using Wikipedia. In their work, WordNet was used in conjunction with the English Wikipedia to generate very high accuracy for polysemous words. Furthermore, there are several works based on computation of similarity between two ontologies, such as[16] in which the authors used the set theoretic Tversky similarity

measure in the matching process. The model proposed focuses on the matching between classes and entities in two ontologies.

[13] proposes a graph-based algorithm for calculating similarity between entities in a graph. This algorithm is based on the Open Directory Project (ODP) which is a large human constructed directory of the Web, using portals and searchengines. One of the models which proposes a new similarity model based on the structure of an ontology is [18]. This paper proposes Ontology Structure based Similarity (OSS) in which an a-priori score is calculated to measure similarity between two concepts in an ontology [9] also concentrated on measuring similarity between two concepts in a single ontology. None of these works tries to model a sentence or a text document by its features such as Wikipedia categories in order to measure semantic similarity between two documents.

Chapter 3

Approach

3.1 Basics

HSA uses Wikipedia's vast knowledge base. Specifically we are spotting Wikipedia resources in the input pair of natural language texts. After spotting, these resources are mapped to Wikipedia articles and the category hierarchy of those articles is fetched to form the category hierarchy of each text input. The intuition is that that text are similar to one another tend to have categories in their individual hierarchies in common. HSA takes advantage of this intuition by using two step method first: It forms the category vector from the category hierarchy and then it compares the vectors together to get a similarity score.

3.1.1 Wikipedia Category Hierarchy Challenges

A challenge in using the Wikipedia category hierarchy lies in the structure of the hierarchy itself. The structure of Wikipedia category hierarchy is a graph. It has a very rich information to mine but it also introduces difficulties in processing it.

Every article in Wikipedia is classified to one or more categories, and can act as its own root for an article based category hierarchy. The article rooted category hierarchy can be constructed by starting with all of the article"s category, then branching from each of those categories to their supercategories to form a category hierarchy for the whole text.

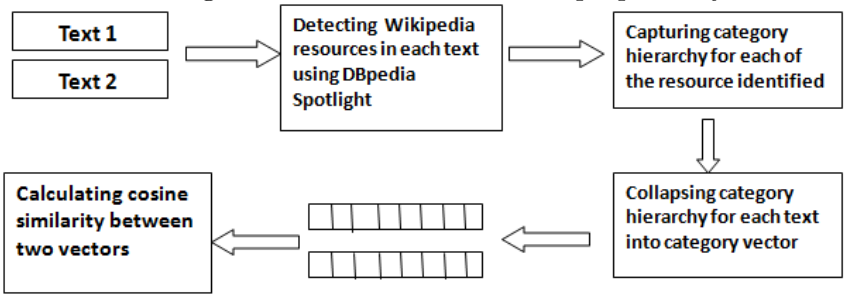
3.2 Hierarchical Semantic Analysis

Our approach is to determine the semantic similarity between two texts by finding Wikipedia resources in texts and then using the commonalities between their Wikipedia category hierarchies as the similarity. Intuition behind this idea is that when two texts have categories in common, even

super categories up several levels, it follows that the articles are similar as they have been tagged to belong to the same category hierarchies.

We chose to use Wikipedia as it is currently the largest knowledge repository on the web. It contains over 4.4 million articles in the English version alone as of March 2014 [1]. Wikipedia like any other taxonomy, contains internal structure classification. These structures include the categories that an article belongs to as well as internal links to other Wikipedia articles that are in the article text. Wikipedia grows incredibly quickly and contains both esoteric and recent concepts. The architecture of proposed system is shown in Figure 1.

Figure 3.1: Architecture of the proposed system



There are three major steps in the HSA.
1) Wikipedia Resource Identification from input text.
2) Capturing and collapsing the category hierarchy into category vector for each text to further calculate the cosine similarity.
3) Calculating the similarity score between two vectors.

3.2.1 Wikipedia Resource Identification

Given an input text, first we intend to identify Wikipedia resources in the text. Resource Identification includes both entity recognition and entity disambiguation i.e. resources recognised in the text are mapped to correct Wikipedia resource(according to context) so that category hierarchy can be fetched for the next step. Both recognition and disambiguation are performed using DBpedia Spotlight[14].It annotates the input text with the Wikipedia resources identified in the text. For the following text fragment, annotations generated by DBpedia Spotlight are shown in braces aside the wikipedia resource identified:
Rahul Dravid(Rahul_Dravid) has been described as one of the greatest batsmen in the history of cricket. He was named one of the best five cricketers of the year by WisdenCricketers (Wisden_Cricketers_Almanack) Almanack in 2000 and received the Player of the Year and the Test Player of the Year awards at the in-

augural ICC(International_Cricket_Council)awards ceremony in 2004.In December 2011, he became the first non-Australian cricketer to address at the BradmanOration(Sir_Donald_Bradman_Oration in Canberra(Canberra). As of October 2012, Dravid(Rahul_Dravid)is the fourth-highest run scorer in Test cricket, after Sachin Tendulkar(Sachin_Tendulkar),Ricky Ponting(Ricky_Ponting) and Kallis(Jacques_Kallis).He is only the second Indian cricketer, after Tendulkar (Sachin_Tendulkar) to score 10,000 runs in ODIs(One_Day_International). Accuracy of concept identification and disambuigation by Spotlight is quite promising.

3.2.2 Capturing and collapsing category hierarchy

After identifying the resources in input text, category hierarchy of each of resource identified is captured from Wikipedia XML dump. But, comparison of two category hierarchies can be accomplished through collapsing the category hierarchy for the text down into a category vector. The category vector contains the categories that represent all of the unique categories that exist in the text's category hierarchy, each mapped to a weighed value for the category.

The specific algorithm used to collapse a text's categories is explained below:

There are three main if else statements in the algorithm:

-The outermost if else statement takes care of the level.The variable level is the shortest path between that category and the base article.

-The inner if else statement takes care of the point that if the category is already visited(that means it is already in the vector because this category lies in category hierarchy of the resource which is already seen) then its weight is updated in the weighmap (hashmap storing the weights of the categories) only if the new weight is greater than the previous one.

-The innermost if else statement is for assigning different weights to categories based on their density. That is if it is a level 2 category, a value based on the number of pages it has(so that value is less for common categories) is added to the weight of the category based on the level and if it is level 3/4/5 category value is decided according to the number of subcategories it has.

The recursive algorithm has two important variables. First, the algorithm allows maximum depth to be specified, the maximum depth is

Algorithm 1: Vectorize Algorithm

Data: level : current depth the algorithm is working at
cat : page or category to get the categories for
wf:weighing function=1/log(level)
np: number of pages of a category
ns: number of subcategories of a category

Result: weigh_map:categories mapped to weighed value

```
if level <=5 then
    if !weighmap.containsKey(cat) then
        if level==2 then
            weighmap.put(cat,wf(level)+(wf(level)*1/np))
        else
            weighmap.put(cat,wf(level)+(wf(level)*1/ns))
    else
        if level==2 then
            if (wf(level)+wf(level)*1/np)>weighmap.get(cat) then
                weighmap.put(cat,wf(level)+wf(level)*1/np)
        else
            if wf(level)+wf(level)*1/ns)>weighmap.get(cat) then
                weighmap.put(cat,wf(level)+wf(level)*1/ns)
    vectorize(cat,level+1)
else
    return
```

the maximum path length from the resource to a category through other categories. The categories near the root of the overall Wikipedia category hierarchy are also general in nature, and thus are not as helpful in determining similarity. For these reasons, it makes sense to limit the articles category hierarchy to a specified maximum depth. Second is the weighting function, wf(level). This function is applied to every category when it is placed into the weighing map.

The weighing function assign weights to the categories in such a way that categories at the same level are not weighed equally. The categories which are dense i.e which contains more subcategories or pages are weighed lesser than the categories which have lesser children. The weighing map includes unique categories. Since Wikipedia allows multiple inheritance and cycles, there exists the possibility that a category is listed twice in an texts hierarchy. In this case, it is necessary to select a single instance of the category to place into the category vector. We chose to use the instance of the category that has the largest weight. Intuitively, it makes sense that having a higher weight for the category gives a more accurate measure of similarity, and so the larger weight is selected.

3.3 Implementation

3.3.1 Wikipedia Database

The Wikipedia database is at the heart of the implementation. Without its vast knowledge store, HSA would not have anything to work on. The version of Wikipedia that HSA is currently running on is a download from March of 2014. Due to the nature of HSA using only the internal structure of Wikipedia rather than any of the content, it was not necessary to download the entirety of the Wikipedia database. We only had to download the tables that directly dealt with the structures we were interested in: Page, Category, PageLinks, CategoryLinks.

3.3.2 Preprocessing

WikiRelate!, ESA and WLM all have detailed preprocessing that must be followed to provision the Wikipedia data for use. One of the initial goals of HSA was to create a system that could be set up quickly and easily from a Wikipedia database dump, without needing preprocessing.
We have created a table category_hierarchy by joining the tables Page and CategoryLinks(which contains three attributes page_id, page_title and parent_id) to facilitate fast and easy access of parent categories while creating the category hierarchies.

3.3.3 HSA

After setting up and preprocessing the database we decided to implement HSA in the Java programming language.There are 3 major steps followed in this thesis to calculate the similarity score between two texts.

Step 1 :Mapping of named entities to correct Wikipedia concept To do this mapping we have used DBpedia Spotlight tool [14]. It works in four stages: The *spotting stage* recognizes in a sentence the phrases that may indicate a mention of a DBpedia resource. It uses the extended set of labels in the lexicalization dataset to create a lexicon for spotting. The implementation used was the LingPipe Exact Dictionary-Based Chunker which relies on the Aho-Corasick string matching algorithm with longest case-insensitive match. Since for many use cases it is unnecessary to annotate common words, a configuration flag can instruct the system to disregard in this stage any spots that are only composed of verbs, adjectives, adverbs and prepositions. The part of speech tagger used was the LingPipe implementation based on Hidden Markov Models.

Candidate selection is subsequently employed to map the spotted phrase to resources that are candidate disambiguations for that phrase.It uses the DBpedia Lexicalization dataset for determining candidate disambiguations for each surface form. The candidate selection offers a chance to narrow down the space of disambiguation possibilities.

The *disambiguation* stage, in turn, uses the context around the spotted phrase to decide for the best choice amongst the candidates. After selecting candidate resources for each surface form, our system uses the context around the surface forms, e.g. paragraphs, as information to find the most likely disambiguations.

The *annotation* can be customized by users to their specific needs through configuration parameters.

Step2: Capturing Category Hierarchy for the concepts identified in Step1: To capture the category hierarchy for a concept, we first query the page table on resource name to get the page id. In the page table, when we are querying for an article page we need to specify the name space as 0 and if we are querying for categories we need to specify the name space as 14.After getting the page id for the resources we are querying the category links table to get the first level categories. After getting the first level categories, each category is queried on category hierarchy table recursively to get the parent categories until the specified level is achieved.

Category hierarchy generated is a DAG.For short sentences,average number of unique categories is 240. For paragraphs,average number of unique categories is 1200.Total height till the root is 5 (We tried with 3,4,5, not on whole data-set but on few pairs, in which 5 was giving best results.Average number of pages per categories is 7200.Average number of sub-categories per category is 3800.

Step 3:Collapsing the category hierarchy into category vector To compare two texts' category hierarchies we need to collapse the category hierarchy into category vector. We have implemented the vectorize algorithm in Java language described in Algorithm 1 to generate category vector for each text.

This algorithm assign weights to the categories based on their level in the hierarchy and based on their density.After identifying Wikipedia resources occurring in the text, their categories are fetched. Each category is weighed according to their level as shown in the table 3.1. If the category is

Table 3.1: Weights assigned by different weighing functions

Level	Weights Assigned by wf=1/level	Weights assigned by wf=1/log(level)
1	1	1
2	0.5	0.73
3	0.3	0.46
4	0.25	0.36
5	0.2	031

already visited(that means it is already in the vector because this category lies in category hierarchy of the resource which is already seen) then its weight is updated in the vector(storing the weights of the categories) only if the new weight is greater than the previous one.

Density factor is also added to these weights which depends on the number of pages the category has if it is a first level category and depends on the number of sub-categories it has if it is a level 3/4/5 category. We are assigning different weights to categories based on their density. That is if it is a level 2 category, a value based on the number of pages it has(so that value is less for common categories) is added to the weight of the category based on the level and if it is level 3/4/5 category value is decided according to the number of sub-categories it has.

Step 4:Calculating the similarity between two category vectors Once a weighed category vector has been formed for both the texts, the vectors can be compared. We have used three vector comparison methods

to compare two category vectors obtained in *Step3*

a)Jaccard Coefficient:

$$\boxed{J(A,B) = \frac{|A \cap B|}{|A \cup B|}} \tag{3.1}$$

The algorithm used to calculate similarity using Jaccard is explained in Algorithm 2.The intersection of two vectors is calculated by taking the average of weighted values of the common categories, rather than picking higher or lower. For union vector, when categories that are in common between the two category vectors are found,the union operation will take the largest weighted value found in the category vectors for that category. Finally, summation of intersection vector over the summation of union vector is returned.

Algorithm 2: Jaccard Based Algorithm

Data: weighmap1: hashmap for category hierarchy of text1
weighmap2: hashmap for category hierarchy of text2
int_map: intersection hashmap
uni_map: union hashmap
cat: category

Result: sim_score: similarity score between two category vectors

```
foreach cat ∈ weighmap1 do
    if weighmap2.contains(cat) then
        int_map.add(cat,(weighmap1.get(cat)+weighmap2.get(cat))/2)
foreach cat ∈ weighmap1 do
    if weighmap2.contains(cat) then
        if weighmap1.get(cat) >weighmap2.get(cat) then
            uni_map.add(cat,(weighmap1.get(cat))
        else
            union_map.add(cat,(weighmap2.get(cat))
sim_score ⟵sum-of-values(int_map)/sum-of-values(uni_map))
return sim_score
```

b)Dice Coefficient:

$$\boxed{D(A,B) = 2\frac{|A \cap B|}{|A| + |B|}} \tag{3.2}$$

The algorithm used to calculate similarity using Dice Coefficient is explained in Algorithm 3. Dice's Coefficient is a simple set based similarity measure that gives a stronger emphasis to common categories in the category vectors due to doubling the weight of the intersection. This emphasis while comparing gives a wider range of scores when differing weights and counts of common categories are found. The formula above works on the input A and B, which are the two category vectors being compared.

Algorithm 3: Dice Based Algorithm

Data: weighmap1: hashmap for category hierarchy of text1
weighmap2: hashmap for category hierarchy of text2
int_map: intersection hashmap
cat: category

Result: sim_score: similarity score between two category vectors

foreach *cat* ∈ *weighmap*1 **do**
  if *weighmap2.contains(cat)* **then**
    int_map.add(cat,(weighmap1.get(cat)+weighmap2.get(cat))/2)

sim_score ⟵2*sum-of-values(int_map)/(sum-of-values(weighmap1)+sum-of-values(weighmap2))

return sim_score

(c) Cosine Measure:

$$\boxed{C(A,B) = \frac{|A|.|B|}{||A||||B||}} \tag{3.3}$$

Cosine similarity is used in many applications, perhaps most often in text mining. It works well to find the similarity between TF-IDF vectors. We selected cosine similarity because it was used in other similarity and relatedness applications, both in those that use Wikipedia like ESA and WLM as well as those that use more traditional taxonomies.
The cosine similarity based algorithm in Algorithm 4 was the most computationally complex algorithm, as the dot product and Euclidean distance for both category vectors needs to be calculated. This creates a much slower overall execution speed than either the Dice or Jaccard methods.
Like the intersection vectors from the Dice and Jaccard algorithms, the individual category vectors are extended out to include the categories from the other category vector. The new categories in the category vectors have a value of 0, and do not affect the dot product or euclidean distance.

The algorithm used to calculate similarity using Cosine measure is explained below:

Algorithm 4: Cosine Based Algorithm

Data: weighmap1: hashmap for category hierarchy of text1
weighmap2: hashmap for category hierarchy of text2
len_map1:length of category vector for text1
len_map2:length of category vector for text2
int_map: intersection hashmap
cat: category
num: numerator for cosine measure

Result: sim_score: similarity score between two category vectors

```
num ⟵0
foreach cat ∈ weighmap1 do
    if weighmap2.contains(cat) then
        num ⟵num + (weighmap1.get(cat)*weighmap2.get(cat))
foreach cat ∈ weighmap1 do
    len_map1 ⟵len_map1 +(weighmap1.get(cat)*weighmap1.get(cat))
len_map1 ⟵square-root(len_map1)
foreach cat ∈ weighmap2 do
    len_map2 ⟵len_map2 +(weighmap2.get(cat)*weighmap2.get(cat))
len_map2 ⟵square-root(len_map2)
sim_score ⟵ num/(len_map1*len_map2)
return sim_score
```

Chapter 4

Empirical Evaluation and Results

Existing similarity measures that use Wikipedia have used the metrics like Pearson's Correlation Coefficient or Spearman's Coefficient to evaluate how accurate the proposed methodology is. These coefficients give values between 0 and 1 that shows how much correlated the benchmark dataset ranking are to the similarity methods rankings.

4.1 DataSets and Evaluation Procedure

Humans have a natural aptitude to judge semantic similarity between texts. Human judgements on a reference set of text pairs can thus be considered correct by definition and as benchmarks against which computer algorithms are evaluated. In this work, we use two such datasets, which are to the best of our knowledge the largest publicly available collections of their kind.

To assess short sentence similarity, we used a collection of 30 sentence pairs which have even distribution of similarity scores from STSS-65 collection,which contains 65 sentence pairs. Each pair is rated by 32 human participants, which were averaged for each pair to produce a single relatedness score. Pearson correlation coefficient was used to compare computed similarity scores with human judgements.

For document similarity, we used a collection of 50 documents from the Australian Broadcasting Corporation's news mail service. A set of 30 document pairs is selected which covers the similarity values from 0 to 1, and each of the pairs has 8-12 human judgements. We used Pearson's linear correlation coefficient to compare computed scores.

4.2 Results

HSA very well captures the semantic similarity of short texts which are not at all syntactically similar. For example *"An Asylum is a psychiatric hospital"* and *"A cemetery is a place where dead people"s bodies or their ashes are buried."* These two sentences do not have a single word in common but HSA says that they have 0.1 semantic relatedness which makes sense as both are places and are related to humans.

HSA also performs well for paragraph pairs as if we consider two paragraphs *"A radical armed Islamist group with ties to Tehran and Baghdad has helped al-Qaida establish an international terrorist training camp in northern Iraq, Kurdish officials say. Intelligence officers in the autonomous Kurdish region of Iraq told the Guardian that the Ansar al-Islam (supporters of Islam) group is harbouring up to 150 al-Qaida members in a string of villages it controls along the Iraq-Iran border. Most of them fled Afghanistan after the US-led offensive, but officials from the Patriotic Union of Kurdistan (PUK), which controls part of north-east Iraq, claim an "abnormal" number of recruits are making their way to the area from Jordan, Syria and Egypt."* and *"Police are combing through videotapes trying to spot the gunman dressed in black who shot a 30-year-old man to death at a downtown massage parlour. The victim was hit in the stomach and upper body and died about 3 and 1/2 hours later in hospital. The woman was not hurt. Police urged business owners to turn over any security-camera videotapes they might have that recorded people on the street at the time. Several such videos are now being reviewed."* In these two paragraphs, there is no syntactic similarity but both the paragraphs are talking about terrorist and some anti-terrorist activities which is captured by HSA giving them the similarity score as 0.3 (same as the human judgement score).

We also tried to examine the algorithm's performance on some random sentences from Wikipedia like: *"A Pirali Brahmin from Calcutta with ancestral gentry roots in Jessore, Tagore wrote poetry as an eight-year-old.[9] At age sixteen, he released his first substantial poems under the pseudonym Bhnusiha ("Sun Lion"), which were seized upon by literary authorities as long-lost classics."* and *"Kipling was one of the most popular writers in England, in both prose and verse, in the late 19th and early 20th centuries.Henry James said: "Kipling strikes me personally as the most complete man of genius (as distinct from fine intelligence) that I have ever known." In 1907, he was awarded the Nobel Prize in Literature, making him the first English-language writer to receive the prize, and to date he remains its youngest recipient."* HSA shows the relatedness of the above sentence pair as 0.3 as the sentences are talking about Rabindranath Tagore and Rudyard Kipling and they both are Indian and English language writers as well as Nobel

Prize laureates.

We varied the weighing function and depth to correlate against to determine the most effective combination of parameters.We first kept weighing function as equivalent to 1/level with the intuition that weight should decrease with the increase in the level but problem with keeping 1/level as the value of weighing function is that it weighs the categories at level 2 as 0.5(out of 1)which is less as far as the importance of the category in contributing to the semantics is concerned. So, we tried with weights provided by 1/log(level)which sounds more reasonable than provided by 1/level as shown in Table 3.1 and results were better by assigning these weights.

We have used the three vector comparison methods to calculate similarity score. Table 4.1 shows the values of correlation coefficient by varying the maximum depth from 3 to 5 when Dice coefficient is used to compare the two vectors. We have created a collection of 30 sentence pairs from STSS-65 since STSS-65 is highly skewed towards the low-similarity end of the scale as a total of 33 out of 65 sentence pairs were rated 0.0 to 0.9 and the rest 32 pairs were rated 1 to 4. So, a subset of 30 pairs containing 10 pairs from 32 pairs and 20 pairs from 32 pairs has been prepared to obtain a more even distribution of across the similarity range based on similar procedure to Islam and Inkpen[5].We have found that the algorithm performed best at maximum depth of 3.

Table 4.1: Correlation by using Dice Coefficient

Dataset	r for level=3	r for level=4	r for level=5
STSS-65	0.64	0.63	0.62
STSS-30	0.903	0.894	0.899

Table 4.2 shows the values of correlation coefficient at three values of maximum depth when Jaccard coefficient is used to compare the two vectors. Here, also we have found that algorithm performed best at maximum depth of 4.

Table 4.2: Correlation by using Jaccard Coefficient

Dataset	r for level=3	r for level=4	r for level=5
STSS-65	0.714	0.721	0.716
STSS-30	0.908	0.912	0.909

Table 4.2 shows the values of correlation coefficient at three values of maximum depth when Cosine measure is used to compare the two vectors. Here, also we have found that algorithm performed best at maximum depth of 4.

Table 4.3: Correlation by using Cosine Similarity

Dataset	r for level=3	r for level=4	r for level=5
STSS-65	0.851	0.833	0.807
STSS-30	0.930	0.911	0.901

From the results we have found the cosine based vector comparison method gave best results with correlation value=0.930 at maximum depth=3 and keeping weighing function as 1/log(n).
Table 1 shows the results of applying our method to estimate relatedness of short sentences. As we can see, HSA technique yield substantial improvements over prior studies. HSA also achieves much better results than the other Wikipedia-based methods. Table 2 shows the results for computing relatedness of entire documents.

Table 4.4: Correlation for STSS Dataset

Author	r (Pearsons Coefficient)
Li et al.	0.816
Kennedy and Szpakowitz	0.873
Kennedy and Szpakowicz	0.851
Feng et al.	0.756
Islam and Inkpen	0.853
OShea et al.	0.838
HSA	0.901

Table 4.5: Correlation for ABC's news mailservice Dataset

Author	r (Pearsons Coefficient)
Bag of words [Lee et al., 2005]	0.1-0.5
LSA[EE et al.,2005]	0.60
ESA-Wikipedia	0.72
ESA-ODP.	0.69
Samer Hassan and Rada Mihalcea	0.684
HSA	0.863

4.2.1 Effect of weighing function

The intuition says that if we use a function that weighs categories that are farther from the concept less, we will achieve better similarity score. We measured the effects of logarithmic and linear functions from shrinking

prespectives on level. Based on the data, logarithmic weighing function yielded better results.

4.2.2 Effect of different depths

Differing the maximum allowable depth also gave variance between correlation coefficients. The value of correlation coefficient when the maximum depth was 3 worked better for STSS dataset but was not promising for ABC's news mailservice dataset. A depth of 5 gave the best correlation coefficients for this dataset.

Chapter 5

Conclusion and Future Work

We proposed a novel approach to compute semantic similarity between natural language texts by using Wikipedia category hierarchy. We use Wikipedia the largest knowledge repository, which contains millions of human-defined concepts. Our approach is called Hierarchical Semantic Analysis, since it uses taxonomy of Wikipedia described by humans. Compared to lexical resources such as WordNet, our methodology has an advantage of using knowledge base that is larger and more comprehensive. Experimental results confirms that using HSA leads to substantial improvements in computing short sentence and paragraph relatedness. Compared with the previous state of the art, using HSA results in notable improvements in correlation of computed similarity scores with human judgements: from r = 0.873 to 0.901 for short sentences and from r = 0.72 to 0.863 for paragraph texts.

The promising results yielded by combining different weighing functions and depth levels can be extended to include different weighing functions or better vector comparison methods. The category hierarchy collapse algorithm can be improved as well in terms of efficiency and processing speed.

Furthermore, in Hierarchical Semantic Analysis, we have only captured the hierarchy of Wikipedia resources found in the text but not the relationship between two resources. We can extend this work to identify relationship between two resources, which if done effectively, can help in understanding the semantics more and can lead to more accurate similarity scores.

HSA can also be used to form clusters of documents in a corpus, each cluster having documents with specified difference in similarity scores.

Bibliography

[1] Wikipedia statistics. `http://en.wikipedia.org/wiki/Wikipedia:Statistics`. Accessed: 2014-03-30.

[2] Jin Feng, Yi-Ming Zhou, and Trevor Martin. Sentence similarity based on relevance. In *In Proceedings of IPMU*. ACM, 2008.

[3] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proc. Of the 20th International joint Conference on Artificial Intelligence(IJCAI07)*, pages 6–12. ACM, 2007.

[4] B. Hajian and T White. Measuring semantic smilarity using a multi-tree model, 2011.

[5] Aminul Islam and Diana Inkpen. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data*, 2(2):1–25, 2008.

[6] J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference on Research in Computational Linguistics*, Taipei, TaiWan, 1997.

[7] Alistair Kennedy and Stan Szpakowicz. Evaluating rogets thesauri. In *Proceedings of ACL-08*, pages 416–424. Association for Computational Linguistics, 2008.

[8] C. Leacock and M. Chodorow. Combining local context and wordnet similarity for word sense identification, wordnet: An electronic lexical database. pages 265–283. MIT Press, 1998.

[9] Wei Nchih Lee, Nigam Shah, Karanjot Sundlass, and Mark Musen. Comparison of ontology-based semanticsimilarity measures. *In AMIA Annual Symposium Proceedings*, page 384, 2008.

[10] Yuhua Li, Zuhair A. Bandar, , and David McLean. An approach for measuring semantic similarity measure between words using multiple information sources. *IEEE transactions on knowledge and data engineering*, 15(4):871–882, 2003.

[11] Yuhua Li, David McLean, Zuhair A. Bandar, James D. OShea, and Keeley Crockett. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering, 2006*, 18:1138–1150, 8 2006.

[12] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, Madison, Wisconsin, USA, 1998.

[13] A.G. Maguitman, F. Menczer, H. Roinestad, and A. Vespignani. Algorithmic detection of semantic similarity. In *Proceedings of the 14th international conference on World Wide Web*, pages 107–116. ACM, 2005.

[14] Pablo N. Mendes, Max Jakob, Andrs Garca-Silva, and Christian Bizer. Dbpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems(I-Semantics)*. ACM, 2011.

[15] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of IJCAI-95*, pages 448–453, Montreal,Canada, 1955. ACM.

[16] M.A. Rodriguez and M.J. Egenhofer. Determining semantic similarity among entity classes from different ontologies. *IEEE transactions on knowledge and data engineering*, pages 442–456, 2003.

[17] M. Ruiz-Casado, E. Alfonseca, and P. Castells. Automatic assignment of wikipedia encyclopedic entries to wordnet synsets. *Advances in Web Intelligence*, pages 380–386, 2005.

[18] V. Schickel-Zuber and B. Faltings. Oss: a semantic similarity function based on hierarchical ontologies. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 551–556. Morgan Kaufmann Publishers Inc., 2007.

[19] Michael Strube and Simone Paolo Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In *proceedings of the 21st national conference on Artificial intelligence*, volume 2, pages 1419–1424. AAAI06, AAAI Press, 2006.

[20] Tversky. Features of similarity. *Psychological Review*, 84:327–352, 1955.

[21] I.H. Witten and D. Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia linkse. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy*, pages 25–30, Chicago,USA, 2008. AAAI Press.

[22] Z. Wu and M. Palmer. Verb semantics and lexical selection. In *Proceedings of 32nd annual Meeting of the Association for Computational Linguistic*, Las Cruces, New Mexico, 1994. Association for Computational Linguistics.

[23] Y. Wang Z. Zhou and J. Gu. New model of semantic similarity measuring in wordnet. In *Proceedings of the 3rd International Conference on Intelligent System and Knowledge Engineering,*, Xiamen, China, 2008.