

Semantic Distances for Sets of Senses and Applications in Word Sense Disambiguation

Dimitrios Mavroeidis, George Tsatsaronis and Michalis Vazirgiannis

Department of Informatics, Athens University of Economics and Business, Athens, Greece
{dmavr,gbt,mvazirg}@aueb.gr

Abstract. There has been an increasing interest both from the Information Retrieval community and the Data Mining community in investigating possible advantages of using Word Sense Disambiguation (WSD) for enhancing semantic information in the Information Retrieval and Data Mining process. Although contradictory results have been reported, there are strong indications that the use of WSD can contribute to the performance of IR and Data Mining algorithms. In this paper we propose two methods for calculating the semantic distance of a set of senses in a hierarchical thesaurus and utilize them for performing unsupervised WSD. Initial experiments have provided us with encouraging results.

1. Introduction

Towards the direction of improving the accuracy in the retrieval process, the information retrieval community has been investigating the possible advantages of using Word Sense Disambiguation (WSD) [1] for enhancing both the query and the content with semantics. In spite of the early discouraging results [2], recent studies have clearly indicated that WSD algorithms achieving an accuracy of 50-60% can improve significantly the precision of IR tasks [3,4]. More precise experimental efforts [5] have even reported an absolute increase of 1.73% and a relative increase of 45.9% in precision whilst utilizing a supervised WSD algorithm that reported an accuracy of 62.1%.

From the Data Mining Community perspective, the process of applying WSD for improving clustering or classification results has produced contradictory results. In [6,7] the results presented were negative, though probably because in [7] the WSD process applied did not assign a single sense to each word, but tackled all the possible senses for all the words, while in [6] the semantic relations, like the hypernym/hyponym relation, were not taken into account. In contrast, in [8,9], a rich representation for senses was utilized, that exploited the semantic relations between senses, as provided by WordNet [10]. Thus, there exist indications that the correct usage of senses can improve accuracy in Data Mining tasks.

In general a WSD process can be either supervised or unsupervised (or a combination of the two). The supervised WSD considers a pre-tagged text corpus that is used as a training set. The sense of a new keyword can then be inferred based on

the hypothesis generated by the training set. A simple supervised learning algorithm is to calculate the frequencies of all the possible senses of a given keyword and assign the most probable sense (naïve bayes classifier). In WordNet [10], the senses for each word are ranked according to a probability distribution found in a large text corpus, thus the assignment of the first sense, as provided by WordNet, to a keyword is equivalent to applying a naïve bayes WSD algorithm. Although supervised approaches seem to outperform unsupervised ones, it can be argued that in specific domains the cost of constructing a training set for training a WSD algorithm can be prohibitive, and thus for such domains an unsupervised WSD algorithm may be more appropriate.

In this paper we propose an unsupervised WSD algorithm that utilizes a background hierarchical thesaurus (a given ontology describing the hypernym/hyponym relation of senses) and WordNet. Firstly, using WordNet the set of all possible senses for a keyword are identified; then, the given ontology is utilized for identifying the correct sense of each keyword, using the notion of compactness. Compactness is used for measuring the level of semantic similarity of a set of senses, in order to choose the ‘best’ set. Our approach follows the intuition that adjacent terms extracted from a given document are expected to be semantically close to each other. We present two methods for retrieving the semantic similarity of a set of senses using a hierarchical thesaurus.

Our first approach is by means of computing a modification of the Steiner Tree [11] of a set of senses and their least common ancestor in the WordNet graph. The Tree is computed with the precondition that every terminal sense has a path to the least common ancestor. The compactness of the set of senses is computed based on the sum of the weights of the edges of this Tree.

The second approach relies on a mapping scheme that maps a given ontology to a vector space; then the structure of the vector space is exploited in order to define a compactness measure by means of the centroid. This approach provides us with a geometrically interpretable compactness measure that evaluates the level of semantic similarity of a set of senses. It can be shown that there exist standard metrics in the ontology and the vector space, such that the mapping is isometric. The compactness of a set of senses is computed by means of the sum of the distances of the vector senses to their centroid.

The rest of the paper is organized as follows. Section 2 discusses the related work concerning unsupervised WSD methods that rely on concept hierarchies. Section 3 presents our first compactness measure that is based on the graph structure of WordNet. Section 4 describes our second compactness measure that relies on a mapping scheme from the ontology to a vector space. Section 5 discusses the experiments performed. Finally section 6 contains the concluding remarks and pointers to further work.

2. Related Work

The exploitation of WordNet as a concept hierarchy has constituted the base of many WSD algorithms, both supervised and unsupervised. In this section we shall briefly

Semantic Distances for Sets of Senses and Applications in Word Sense Disambiguation

describe the relevant work done in unsupervised WSD algorithms utilizing WordNet as their concept hierarchy. Sussna [12] proposes a disambiguation algorithm, which assigns a sense to each noun in a window of context by minimizing a semantic distance function among their possible senses. The measure proposed is based on the assignment of weights to the edges in the WordNet noun hierarchy. For the weights computation, the *is-a*, *has-part*, *is-a-part-of* and *antonyms* relations between the noun senses are considered. Furthermore, the higher the level of the WordNet hierarchy, the greater is the conceptual distance that a semantic link between two senses suggests. Thus, Sussna's algorithm rewards best semantic links between senses existing low in the WordNet noun hierarchy, which is rational, since the lower the level in the WordNet hierarchy of a given link, the higher the conceptual connection between the two linked specialized (due to their depth) senses. Besides the fact that this proposed method has combinatory complexity due to the pair-wise computation of the semantic distance function for a given window of context, the conceptual density of the window available senses is not computed as a whole, but as a sum of pair-wise semantic distances.

Aggire and Rigau [13] introduce and apply a similarity measure based on conceptual density between noun senses. Their proposed measure is based on the *is-a* hierarchy in WordNet and it measures the similarity between a target noun sense and the nouns in the surrounding context. For this purpose, they divide the WordNet noun *is-a* hierarchy into subhierarchies, where each possible sense of the ambiguous noun belongs to a subhierarchy. The conceptual density for each subhierarchy describes the amount of space occupied by the nouns that occur within the context of the ambiguous noun. This actually measures the degree of similarity between the context and the possible senses of the word. For each possible sense the measure returns the ratio of the area occupied by the subhierarchies of each of the context words within the subhierarchy of the sense to the total area occupied by the subhierarchy of the sense. The sense with the highest conceptual density is assigned to the target word.

Banerjee and Pedersen [14] suggest an adaptation of the original Lesk algorithm in order to take advantage of the network of relations provided in WordNet. Rather than simply considering the glosses of the surrounding words in the sentence, the concept network of WordNet is exploited to allow for glosses of word senses related to the words in the context to be compared as well. Essentially, the glosses of surrounding words in the text are expanded to include glosses of those words to which they are related through relations in WordNet. They also suggest a scoring scheme such that a match of n consecutive words in the glosses is weighted more heavily than a set of n one word matches.

In order to clarify the notion of semantic distance we will review the most popular semantic distances defined for ontologies. Before we review the most popular semantic distances for ontologies it is necessary to present the definition of path size in an ontology.

Definition 1[Path size]: Let O be an ontology and $p=(v_1, \dots, v_n)$ be a path in the ontology from sense v_1 to sense v_n that is defined by n vertices of the ontology. We will call the size of the path, $size(p)$ as the sum of the weight of all the edges that are contained in the path.

A common element of almost all the semantic similarity measures on IS-A relations is that the similarity of two concepts c_1 and c_2 depends on the size of the

shortest path from c_1 to c_2 , that goes through a common ancestor of c_1 and c_2 . More precisely, the larger the size of the shortest path from c_1 to c_2 , the larger the semantic distance. If the least common ancestor of c_1 and c_2 , $lca(c_1, c_2)$ exists then the shortest path can be encoded as the path from c_1 to $lca(c_1, c_2)$ and from $lca(c_1, c_2)$ to c_2 . The only exception is the Resnik measure, where the similarity between two concepts is depends on the size of the path form the root of the ontology to the least common ancestor.

Resnik Measure [15]: The similarity of two concepts c_1 and c_2 that lie in an IS-A ontology is defined as:

$$Sim(c_1, c_2) = \max_{c \in Supp(c_1, c_2)} IC(c)$$

Where $IC(c)$ is the information content of concept c and where $Supp(c_1, c_2)$ represents all the concepts in the ontology that are more general than both c_1 and c_2 .

Hirst-St-Onge Measure [16]: The strength of the relationship between two concepts c_1 and c_2 in an ontology is defined as:

$$Rel(c_1, c_2) = C - pathlength - k \cdot d$$

Where d is the number of changes in the direction of the path and C and k are constants.

Jian-Conrath Measure [17]: The distance of two concepts c_1 and c_2 that lie in an IS-A ontology is defined as:

$$Dist(c_1, c_2) = IC(c_1) + IC(c_2) - 2 \cdot IC(lca(c_1, c_2))$$

Where $IC(c)$ is the information content of a concept c and $lca(c_1, c_2)$ is the lowest common ancestor of c_1 and c_2 .

Leacock-Chodorow Measure [18]: The similarity of two concepts c_1 and c_2 that lie in an IS-A ontology is defined as:

$$Sim(c_1, c_2) = -\log\left(\frac{len(c_1, c_2)}{2D}\right)$$

Where len is the length of the path that connects the two concepts and D denotes the maximum depth of the taxonomy.

Lin similarity [19]: The similarity of two concepts c_1 and c_2 that lie in an IS-A ontology is defined as:

$$Sim(c_1, c_2) = \frac{2 \cdot IC(lca(c_1, c_2))}{IC(c_1) + IC(c_2)}$$

Where $lca(c_1, c_2)$ defines the least common ancestor of concepts c_1 and c_2 .

We can easily observe that the common characteristic of these measures is that the semantic distance of two senses c_1 and c_2 depends on the size of the shortest path (through the least common ancestor) that connects c_1 and c_2 in the ontology, or on the size of the path that connects the $lca(c_1, c_2)$ to the root of the ontology.

3. Compactness Measure Based on Ontology Graph

As we have discussed in the introduction section we aim in computing the semantic similarity of a set of concepts. Semantic similarity between two concepts that reside in an ontology depends on the shortest path (through a least common ancestor) that connects the two concepts. Thus, it is natural to investigate possible extensions of the shortest path notion for a set of concepts. In order to define the compactness measure we will look into graph theoretic measures that account for the shortest path that connects a set of concepts.

The Steiner Tree, defined as the Tree with the smallest cost that contains a set of concepts, is the graph theoretic notion that we will utilize in order to define our compactness measure. More precisely we will use a modification of the Steiner Tree that takes into account the nature of semantic similarities in an ontology. Recall that the distance between two concepts in an ontology is not defined as the shortest path that connects them in the ontology, but rather as the shortest path that goes through a common ancestor. Thus, it can be argued that two concepts are connected only through a common ancestor and not through any other path in the ontology. Consequently, it is natural to consider the computation of the Steiner Tree of the set of concepts, with their least common ancestor such that each concept has one path to the least common ancestor. The existence of the least common ancestor (and of a path of every concept to the least common ancestor) would guarantee that a path connecting all pairs of concepts (in the context discussed earlier) exists in the Steiner Tree.

We can now define the common distance of a set of concepts as the cost of the Steiner Tree of the concepts and their least common ancestors such that each concept has one path to the least common ancestor. Under this definition, the most compact set of concepts will be the set with the smallest common distance.

4. Compactness Measure Based on Mapping

In this section we will present the details of the mapping from the ontology to a vector space. As our main aim is to introduce a compactness measure, we are interested in preserving our ability to measure semantic distances in the new vector space.

4.1 Mapping of tree ontology to vector space

When mapping an ontology to a new space (a vector space) we aim in preserving our ability to measure semantic distances (as we can in the ontology by means of the least common ancestor etc.) in the new space. The mapping of the ontology to a vector space would provide us with the capability of using geometrically interpretable compactness measures (such as by means of the centroid).

We will now present the main notions and definitions for mapping the concepts from the tree ontology to a vector space. The structure of the vector space would

allow for the calculation of a geometrically interpretable compactness measure with the use of the centroid. Since the compactness measure that we aim to use in the vector space relies on the distances between the vectors, our main goal in the mapping procedure will be preserve our ability to measure semantic distances on the vector space using standard vector space distances (such as the Euclidean, Manhattan, etc.). We will firstly define the vector space on which the concepts will reside.

Definition 2: Let O be an IS-A tree ontology. We define the Ontology vector space V_O , an n -dimensional real vector space (where n is the number of edges of the tree ontology), where each edge of the ontology corresponds to a dimension of V through the function $corr(i,j)$. $corr(i,j)$ denotes that the i^{th} dimension corresponds to the j^{th} edge.

Now we will define the exact process with which the concepts in the ontology are mapped to the vector space.

Definition 3: Let O be an IS-A tree ontology and V be a vector space, we define a function from the Ontology O to V as $f_g : O \rightarrow V$ with $f_g(c) = (x_1, \dots, x_n)$. If (e_1, \dots, e_k) are the weights corresponding to the edges of the path of c to the root, and $corr(i,j)$ denotes the correspondence of the i^{th} dimension to the j^{th} edge, we will have that $x_i = g(e_j)$, where g is a function that maps edge weights to the corresponding dimensions. For the remaining x_j (no edge correspond to these x_j) we will have $x_j = 0$. We will refer to the $f_g(c)$ as concept vectors.

As we have discussed in the Related Work section the semantic distances that are defined on an IS-A ontology depend on the size of the shortest path from c_1 to c_2 through a common ancestor, or on the size of the path from the least common ancestor of c_1 and c_2 to the root of the ontology. In the following propositions we will show that there exist mappings f_g and standard distance and similarity measures in V_O vector space (i.e. product, the Euclidean distance) such that the distance of two vector concepts $f_g(c_1)$ and $f_g(c_2)$ depends on the size of the shortest path from c_1 to c_2 or on the size of the least common ancestor of c_1 and c_2 to the root of the ontology. We will also show that in the cases of Resnik measure and Jian-Conrath measure there exist mappings and measures in the vector space that produce exactly the same results.

Proposition 1: Let O be an IS-A tree ontology and V_O be the vector space that corresponds to O , then there exists a mapping function f_g such that the inner product of two concept vectors c_1 and c_2 in V_O $\langle f_g(c_1), f_g(c_2) \rangle$ is equal to the size of the path of $lca(c_1, c_2)$ to the root.

Proof.

We consider the mapping f_g from the Ontology to V_O such that for the weights of the edges e_j we will have $g(e_j) = \sqrt{e_j}$

The dot product in vector spaces is defined as:

$$\langle f_g(c_1), f_g(c_2) \rangle = \sum x_i y_i, \text{ where } x_i \text{ and } y_i \text{ are the coordinates of } f_g(c_1) \text{ and } f_g(c_2).$$

From the embedding procedure we will have that the vector concepts will have common coordinates for the dimensions that correspond to the path from $lca(c_1, c_2)$ to the root, and for all the other dimension it will be either be $x_i = 0$ or $y_i = 0$.

Thus, we can write:

$$\langle f_g(c_1), f_g(c_2) \rangle = \sum x_i^2, \text{ where the } x_i \text{ are the dimensions that correspond to the edges of the path from the } lca(c_1, c_2) \text{ to the root.}$$

Semantic Distances for Sets of Senses and Applications in Word Sense Disambiguation

From the embedding procedure these x_i will correspond to the weights of the edges of the path from the $lca(c_1, c_2)$ to the root. Thus if we have e_i to be the weights of the edges of the path from the $lca(c_1, c_2)$ to the root of the ontology we will have $g(e_i) = \sqrt{e_i}$ and thus we can write:

$\langle f_g(c_1), f_g(c_2) \rangle = \sum e_i$, where the e_i are the weights of the edges that belong to the path from the least common ancestor to the root of the ontology.

Thus we have shown that there exists a mapping $f_g : O \rightarrow V_O$ such that the inner product in V_O is equal to the size of path from the least common ancestor to the root of the ontology. □

As a special case of this proposition we can show that there exists a mapping f_g such that the inner product is equal to the Resnik similarity measure:

Proposition 2: Let O be an IS-A tree ontology, then there exists a function f_g , and a weighting scheme for the ontology, such that the Resnik similarity measure is equal to the dot product in the V_O

Proof.

The weighting scheme that we consider for an edge (v_1, v_2) , where v_1 is more general than v_2 is $IC(v_2) - IC(v_1)$. We consider now a function f_g such that $g(e_i) = \sqrt{e_i}$.

In V_O the dot product will be:

$$\langle f_g(c_1), f_g(c_2) \rangle = \sum x_i y_i$$

From the embedding procedure we will have that the vector concepts will have common coordinates for the dimensions that correspond to the path from $lca(c_1, c_2)$ to the root, and for all the other dimension it will be either be $x_i = 0$ or $y_i = 0$. Thus we can write:

$$\langle f_g(c_1), f_g(c_2) \rangle = IC(lca(c_1, c_2)) - IC(father(lca(c_1, c_2))) + \dots + IC(child(root)) - IC(root) = IC(lca(c_1, c_2)) - IC(root) = (we can assume that the probability of the root is 1 and thus its information content is 0 and write that)$$

$$\langle f_g(c_1), f_g(c_2) \rangle = IC(lca(c_1, c_2))$$

Thus we have shown that there exists a function f_g , and a weighting scheme for the ontology, such that the inner product of the V_O vector space is equal to the Resnik measure. □

Proposition 3: Let O be an IS-A tree ontology and V_O be the vector space that corresponds to O , then there exists a function f_g for each Minkowski distance, such that each Minkowski distance of two concept vectors c_1 and c_2 in V_O is proportional to the shortest path between c_1 and c_2 in the ontology.

Proof.

We consider the mapping f_g from the Ontology to V_O such that for the weights of the edges e_j we will have $g(e_j) = \sqrt[p]{e_j}$

For c_1 and c_2 concepts we will have that the Minkowski distance in V_O is:

$$d^p(f_g(c_1), f_g(c_2)) = \sum |x_i - y_i|^p, \text{ where } x_i \text{ will be the coordinates that correspond to vector } f_g(c_1) \text{ and } y_i \text{ will be the coordinates that correspond to } f_g(c_2). \text{ For the dimensions that correspond to edges that are above } lca(c_1, c_2) \text{ we will have } x_i = y_i. \text{ For}$$

the dimensions that correspond to edges that don't belong to the path of either c_1 or c_2 we will have $x_i = y_i = 0$.

Thus one can write:

$$d^p(f_g(c_1), f_g(c_2)) = \sum_{i \in \text{edges}(c_1, \text{lca}(c_1, c_2))} |x_i|^p + \sum_{i \in \text{edges}(c_2, \text{lca}(c_1, c_2))} |y_i|^p$$

From the definition of the g function we can derive that

$d^p(f_g(c_1), f_g(c_2))$ is equal to the sum of weights of the edges of the shortest path from c_1 to c_2 .

Thus we have shown that for any Minkowski distance, there exists an f_g such that the Minkowski distance on vector space depends on the size of the shortest path in the ontology.

□

For a special case of the above proposition (Minkowski distance with $p=1$), we will show that there exists a mapping such that it is equal to the Jian-Conrath measure.

Proposition 4: Let O be an IS-A tree ontology and V_O the vector space that corresponds to O , then there exist a function f_g such that the Manhattan distance (Minkowski distance with $p=1$) of the V_O vector space is equal to the Jian-Conrath measure.

Proof.

It can be derived in a straight forward manner from proposition 3.

□

In this subsection we have described our mapping scheme that embeds the concepts of an ontology in a vector space such that we preserve the ability to measure semantic distances in the new space. The ability to measure semantic distances is verified by the four propositions presented in this subsection. In the following subsection we will exploit the structure of the vector space in order to define a geometrically interpretable compactness measure with the use of the centroid.

4.2 Compactness Measure

A widely used method for measuring the compactness in vector spaces is by means of the centroid.

$$\bar{c} = \frac{1}{n} \sum \bar{x}_i$$

We will utilize the centroid in order to define a common distance measure for a set of vectors, that measures the density of the senses in the vector space. More precisely the measure of common distance of a set of vectors is defined as the sum of squared distances of the vectors to the centroid, and the most compact set of vectors will be the set with the smallest common distance.

$$cd(x_i) = \sum d^2(\bar{x}_i, \bar{c})$$

4.3 Mapping of non-tree ontologies to vector spaces

As it is clearly indicated, the theory discussed in the previous section applies only to tree ontologies. However, the most widely used ontology WordNet, is not a Tree, and a concept in WordNet can have multiple paths to the root of the ontology. The existence of multiple paths to the root, would prevent us from performing the embedding procedure described in the previous section. In order to overcome this problem, we will consider multiple versions of a concept in the vector space. Each version, will correspond to a distinct path of the concept to the root of the ontology. It can be easily observed that not all distances in the vector space are “valid”. They are not “valid” in the sense that they do not correspond to distances in the original ontology (i.e. multiple version of a concept will have non zero distance). Thus, in order to construct the centroid for the set of vector concepts, we need to consider only the “valid” distances in the vector space.

In order to address the problem we connect all the nodes that correspond to “valid” distances in the original ontology. This can be considered as a graph, where edges correspond to the “valid” distances. Then we can define the centroid of this structure as the centroid of the edges’ centriod. The centroid defined will only consider the “valid” distances in the vector spaces and thus, it can be utilized in order to define a compactness measure in a similar manner as in the previous subsection.

5. Experiments

In this section a series of initial experiments is described with respect to the application of the proposed compactness measure, that relies on the graph structure of the ontology presented in section 3, in WSD. For the purposes of our experiments we used WordNet 1.7.1 [10] for our concept hierarchy and a set of texts semantically annotated with this version of WordNet. The set we used is SemCor 1.7.1, downloadable from [20], which is a subset of the Brown corpus. SemCor 1.7.1 contains 186 semantically tagged Brown Corpus files, with all content words tagged with WordNet 1.7.1 senses, and 166 semantically tagged Brown Corpus files with only the verbs being tagged. In our experiments we only considered nouns, and from the 186 files, we chose the same 4 files that were chosen in [21] so as our experiments could be comparable with the unsupervised WSD method proposed by Agirre and Rigau.

The chosen files were *br-a01* (‘a’ standing for the genre “*Press:Reportage*”), *br-b20* (‘b’ standing for the genre “*Press:Editorial*”), *br-j09* (‘j’ standing for the genre “*Learned:Science*”) and *br-r05* (‘r’ standing for the genre “*Humour*”). The distribution of the nouns contained among the 4 texts, that were semantically tagged with senses contained in WordNet 1.7.1, is presented in Table 1.

Table 1. Distribution of nouns contained in WordNet 1.7.1 among the 4 texts

Text	Nouns contained in WordNet 1.7.1
br-a01	485
br-b20	387
br-j09	621
br-r05	446
Total	1939

5.1 Experimentation Setup

In the series of experiments that follow, we utilized the hypernym-hyponym relation as the link among the WordNet senses and in order to disambiguate the noun words we examined them by taking *windows* of adjacent words, and then finding the most compact set of senses representing the words. By selecting the words to be disambiguated in *windows* (i.e. a window of n words is a set of words containing n adjacent words in a text) we came upon two different problems. The first problem concerned computational complexity. By taking large size windows (i.e. windows of 20 words), if we examined all the possible combinations of the senses corresponding to these words, we would need to examine combinations in the order of magnitude of hundred millions. We tackled this problem by using simulated annealing [22], thus cutting down the number of examined combinations in the order of magnitude of few thousands at most for each window. The second problem regarded with the fact that based on the hypernym-hyponym relation, WordNet 1.7.1 contains 9 different disconnected noun senses hierarchies. It could be the case that in a single senses combination of a window of n words, the senses in that combination are not distributed in the same WordNet hierarchy, thus not allowing for the compactness computation. That problem was partially tackled by considering the compactness in that case as the sum of the individual compactness. Even under that consideration for the compactness computation, there could be cases where in a given combination of senses, a sense is found alone (without any other senses of that combination) in a WordNet hierarchy. In this case compactness measure could not be applied, since compactness could hold in cases where at least two senses are connected with the hypernym-hyponym relation, which can be translated in at least two senses existing in the same WordNet hierarchy. In order to tackle with those cases, we conducted experiments with large size windows (i.e. windows of 20 and 30 noun words), thus increasing the probability that each sense in a given combination exists with at least one more sense in the same WordNet hierarchy. Even by increasing the size of the window though, there were cases where senses in a given combination existed as singles in a WordNet hierarchy. Since the window increment could not solve this problem fully, we decided that each such sense contributed a zero (0) in the total compactness.

For the purposes of our experiments, and in order to evaluate the behavior of the compactness measure in the all the aforementioned situations, we conducted three series of experiments. In all three series, simulated annealing was applied. In the first

Semantic Distances for Sets of Senses and Applications in Word Sense Disambiguation

series, we executed WSD for window sizes varying from 3 to 5, without allowing in any given combination of senses a sense being found alone in a WordNet noun hierarchy. In the second series of our experiments we conducted WSD for window sizes varying from 3 to 10, with the difference from the previous series of experiments being the permission of existence of at most one sense being left alone in a WordNet noun hierarchy, in any given combination of senses. The third series of experiments was executed in large size windows (noun word windows of size 20 and 30) where intuitively we believed our compactness measure would reach its top performance for large coverage. In this last series, we allowed all scenarios in any given combination of senses. In the following section the experiments results are presented and discussed.

5.2 Experiments Results and Evaluation

In Table 2, the results from the first series of the experiments are presented.

Table 2. Precision for the first series of experiments, for window sizes 3-5

Window Size	Disambiguated Nouns	Coverage	Ambiguous Nouns	Precision
3	177	9,12%	61	88,13%
4	113	5,82%	35	87,61%
5	64	3,3%	21	87,50%

From the results Table 2, it is obvious that prohibiting the existence of senses left alone in a WordNet hierarchy given a senses combination, cannot provide us with high coverage. The precision in this low coverage is naturally high. The observation that aroused from this series of experiments is that our compactness measure behaves well when applied in WSD, but higher coverage was needed to document upon this. When trying to increase the window size in this first series of experiments (i.e. windows of size 6 and above), the coverage seemed to decrease. Thus, we conducted the second series of experiments, the results of which are presented in Table 3, where we allowed at most one sense being left alone in a WordNet noun hierarchy, at any given combination of senses.

Table 3. Precision for the second series of experiments, for window sizes 3-10

Window Size	Disambiguated Nouns	Coverage	Ambiguous Nouns	Precision
3	744	38,37%	413	74,46%
4	361	18,61%	168	78,67%
5	219	11,29%	99	75,34%
6	177	9,12%	93	77,4%
7	128	6,6%	61	76,56%
8	83	4,28%	37	75,9%
9	54	2,78%	23	81,48%
10	41	2,11%	16	82,92%

In this second series of experiments, we managed to increase the coverage, while maintaining the precision at high levels. The permission of existence of at most one sense being left alone in a WordNet noun hierarchy did not seem to affect our precision much, which is rational, especially in the medium size windows (i.e. windows of size 9 and 10). This second series of experiments proved encouraging, thus we finally conducted a third series of experiment, where any scenario with regards to senses left alone in a WordNet noun hierarchy would be allowed. Intuitively, this would make sense if we incremented the window size, thus increasing the probability noun senses in any given combination existed with at least one more sense of this window in the same WordNet hierarchy. The results of this final series of experiments are presented in Table 4.

Table 4. Precision for the third series of experiments, for window sizes 20 and 30.

Window Size	Disambiguated Nouns	Coverage	Ambiguous Nouns	Accuracy	
				Compactness	C. Density
20	1939	100%	1371	56,73%	60,1%
30	1939	100%	1371	61,06%	60,1%

The results of this experiment were now comparable with the C. Density measure of Aggire and Rigau [21], since full coverage was reported. Compactness precision was close to C. Density when a window size of 20 was selected, while behaved better than C. Density in a window of 30 words. These initial three series of experiments prove that the proposed compactness measure can be successfully applied in WSD tasks, while, in parallel, we intent to scale our experiments in larger windows and in more SemCor 1.7.1 documents.

6. Conclusions and Further Work

In this paper we have presented two compactness measures for calculating the similarity of a set of concepts that reside in a hierarchical ontology. The first measure relies on the graph theoretic notion of Steiner Tree, while the other relies on the mapping of the ontology concepts to a vector space. We have conducted initial experiments in order to verify our approach for Word Sense Disambiguation, using the graph theoretic compactness measure and SemCor1.7.1. The experiments have produced encouraging results, regarding the ability of our compactness measures to perform Word Sense Disambiguation.

Concerning further work we aim in conducting exhaustive experiments on SemCor and SenSeval [23] datasets, comparing both our approaches to other unsupervised learning algorithms for WSD. Moreover, we aim to investigate possible solutions (such as with gloss overlaps) to overcome the problem of the existence of the 9 disconnected WordNet noun hierarchies.

References

1. Ide, N., Véronis, J.: Word Sense Disambiguation: The State of the Art. *Journal of Computational Linguistics* (1998) 24(1) 1-40
2. Sanderson, M.: Word Sense Disambiguation and Information Retrieval. In: *Proc. of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval* (1994) 49-57
3. Shutze, H., Pederson, J.O.: Information Retrieval Based on Word Senses. In: *Proc. Of the 4th Annual Symposium on Document Analysis and Information Retrieval* (1995) 161-175
4. Gonzalo, J., Verdejo, F., Chugur, I., Cigarran, J.: Indexing with WordNet synsets can Improve Information Retrieval. In: *Proc. of the COLING/ACL'98 Workshop on Usage of WordNet for NLP* (1998)
5. Stokoe, C., Oakes, M.P., Tait, J.: Word Sense Disambiguation in Information Retrieval Revisited. In: *Proc. of SIGIR-03, 26th ACM International Conference on Research and Development in Information Retrieval* (2003) 159-166
6. Kehagias, A., Petridis, V., Kaburlasos, V.G., Fragkou, P.: A comparison of Word- and Sense-based Text Categorization Using Several Classification Algorithms. *Journal of Intelligent Information Systems* (2003) 21(3) 227-247
7. Scott, S., Matwin, S.: Feature Engineering for Text Classification. In: *Proc. of of ICML-99, 16th International Conference on Machine Learning* (1999) 379-388
8. Hohto, A., Staab, S., Stumme, G.: WordNet improves Text Document Clustering. In: *Proc. of the SIGIR 2003 Semantic Web Workshop* (2003)
9. Bloehdorn, S., Hotho, A.: Boosting for Text Classification with Semantic Features. In: *Proc. of the SIGKDD 2004 MSW Workshop* (2004)
10. Website: WordNet – a lexical database for the English Language. <http://www.cogsci.princeton.edu/~wn/>
11. Hwang, R., Richards, D., Winter, P.: The Steiner Tree Problem. In: volume 53 of *Annals of Discrete Mathematics* (1992)
12. Sussna, M.: Word Sense Disambiguation for free-text indexing using a massive semantic network.. In: *Proc. of the second international conference on Information and Knowledge Management* (1993) 67-74
13. Agirre, E., Rigau, G.: A proposal for word sense disambiguation using conceptual distance. In: *Proc. of the 1st International Conference on Recent Advances in NLP* (1995)
14. Banjeree, S., Pedersen T.: Extended gloss overlaps as a measure of semantic relatedness. In: *Proc. of the Eighteenth International Joint Conference on Artificial Intelligence* (2003) 805-810
15. Resnik, P.: WordNet and class-based probabilities. In: C. Fellbaum (Ed.), *WordNet: An electronic lexical database*. MIT Press, (1998) 239-263
16. Hirst, G., Onge, D. St.: Lexical chains as representations of context for the detection and correction of malapropisms. In: C. Fellbaum (Ed.), *WordNet: An electronic lexical database*. MIT Press, (1998) 305-332
17. Jiang, J., Conrath, D.: Semantic similarity based on corpus statistics and lexical taxonomy. In: *Proc. of International Conference on Research in Computational Linguistics* (1997) 19-33.
18. Leacock, C., Chodorow, M.: Combining local context and WordNet similarity for word sense identification. In: C. Fellbaum (Ed.), *WordNet: An electronic lexical database*. MIT Press, (1998) 265-283
19. Lin, D.: Using syntactic dependency as a local context to resolve word sense ambiguity. In: *Proc. of the 35th Annual Meeting of the Association for Computational Linguistics* (1997) 64-71

20. Website: UNT Center for Research on Language and Information Technologies
<http://mira.csci.unt.edu/downloads.html>
21. Agirre, E., Rigau, G.: Word Sense Disambiguation Using Conceptual Density. In: Proc. of COLING-96 (1996) 16-22
22. Cowie, J., Guthrie, J., Guthrie, L.: Lexical disambiguation using simulated annealing. In: Proc. of the 14th International Conference on Computational Linguistics (1992) 359-365
23. Website: Senseval Web page, <http://www.senseval.org>