Comparing Similarity Measures for Original WSD Lesk Algorithm

Sulema Torres and Alexander Gelbukh

Centro de Investigación en Computación (CIC-IPN),
Unidad Profesional Adolfo-López Mateos, Av. Juan de Dios Bátiz s/n
and M. Othón de Mendizábal, Zacatenco, México, DF. 07738, Mexico
sulema7@gmail.com, www.gelbukh.com

Abstract. There are many similarity measures to determine the similarity relatedness between two words. Measures of similarity or relatedness are used in such applications as word sense disambiguation. One of the methods used to resolve WSD is the Lesk algorithm. The performance of this algorithm is connected with the similarity relatedness between all words in the text, i.e the success rate of WSD should increase as the similarity measure's performance gets better. This paper presents a comparison of several similarity measures applied to WSD using the original Lesk Algorithm**.**

Keywords: Word Sense Disambiguation, Lesk Algorithm, WordNet, Semantic Similarity.

1 Introduction

The need to determine the *degree of semantic similarity*, or *relatedness*, between two words is an important problem in Natural Language Processing (NLP). Similarity measures are used in such applications as word sense disambiguation, determining discourse structure, text summarization and annotation, information extraction and retrieval, automatic indexing, lexical selection, and automatic correction of word errors in text [4].

Human beings have an innate ability to tell if one word is more similar to a given word than another. For example, most would agree that the automotive senses of *car* and *tire* are related while *car* and *tree* are not.

There are mainly two approaches to semantic similarity [2, 17]. First approach is making use of a large corpus and gathering statistical data from this corpus to estimate a score of semantic similarity. Second approach makes use of the relations and the hierarchy of a thesaurus, which is generally a hand-crafted lexical database such as WordNet [5]. As in many other NLP studies, hybrid approaches that make benefit from both techniques also exist in semantic similarity.

There are some ways to evaluate semantic similarity measures. One is checking the correlation between the results of similarity measures and human judgments. Another one is to select an application area of semantic similarity, and compare the results of different similarity measure according to the success rates in that

Advances in Computer Science and Applications.
Research in Computing Science 43, 2009, pp. 155-166

application area. In this paper we compare different similarity measures, applying them to WSD.

WSD is one of the most important NLP tasks, and can be defined as the task of automatically assigning the most appropriate sense to a word within a given context. The set of candidate senses are generally available from a lexical database.

Various approaches to word sense disambiguation have been proposed. This paper is based on the so-called **Lesk algorithm** [10].

The paper is organized as follows. We first discuss the related work in Section 2 followed by a description of the Lesk algorithm as a method to resolve WSD in Section 3. Section 4 describes the lexical resource used Wordnet. Next, we present the measures of word semantic similarity used in this work for WSD based on the Lesk algorithm. Then we describe our experimental methodology and results. In Section 8 we present a discussion of the results, and give conclusions in Section 9.

2 Related Work

Many methods in WSD and similar tasks are based on optimization of some word relatedness measure, which gives a numerical estimate of the probability of two words (or word senses) to appear in the same text fragment [7, 6, 10]; the senses are chosen that are more probable in a given context.

Budanitsky and Hirst [4] have compared five different proposed measures of similarity or semantic distance in WordNet: Hirst–St-Onge [7], Leacock–Chodorow [9], Resnik [16], Jiang–Conrath [8] and Lin [11] were experimentally compared by examining their performance in a real-word spelling correction system, specifically, malapropism detection. In their malapropism corrector, words are (crudely) disambiguated where possible by accepting senses that are semantically related to possible senses of other nearby words. They found that there are considerable differences in the performance of five proposed measures of semantic relatedness and Jiang and Conrath's measure was shown to be best overall.

Padwardhan *et al.* [14] generalizes the Adapted Lesk Algorithm of Banerjee and Pedersen [3] to a method of word sense disambiguation based on semantic relatedness and they evaluate a variety of measures of semantic relatedness. These include measures by Lesk [10], Resnik [16], Jiang and Conrath [8], Lin [11], Leacock and Chodorow [9], and Hirst and St. Onge [7]. Then found that the two most accurate methods in their study were quite dissimilar. Adapted Lesk gloss overlaps are based on the definitions found within WordNet, while the measure of Jiang–Conrath is based on the concept hierarchy of WordNet and corpus statistics. This suggests that some combination of gloss overlaps, information content, and path lengths might result in improved accuracy.

Sinha and Mihalcea [18] describe an unsupervised graph-based method for word sense disambiguation, and presents comparative evaluations using several measures of word semantic similarity. They proposes a combination of similarity measures given by a graph where they use the similarity metric *jcn* to draw similarity values between nouns and the similarity metric *lch* to draw similarity values between verbs.

All the other edges in the graph, including links between adjectives and adverbs, or links across different parts-of-speech, are drawn using the *lesk* measure. The results indicate that the right combination of similarity metrics can lead to a performance competing with the state-of-the-art in unsupervised word sense disambiguation.

3 WSD using the Lesk Algorithm

The Lesk algorithm [10] uses dictionary definitions (gloss) to disambiguate a polysemous word in a sentence context. The major objective of his idea is to count the number of words that are shared between two glosses. The more overlapping the words, the more related the senses are.

To disambiguate a word, the gloss of each of its senses is compared to the glosses of every other word in a phrase. A word is assigned to the sense whose gloss shares the largest number of words in common with the glosses of the other words. Figure 1 shows the graphic representation of the Lesk Algorithm.

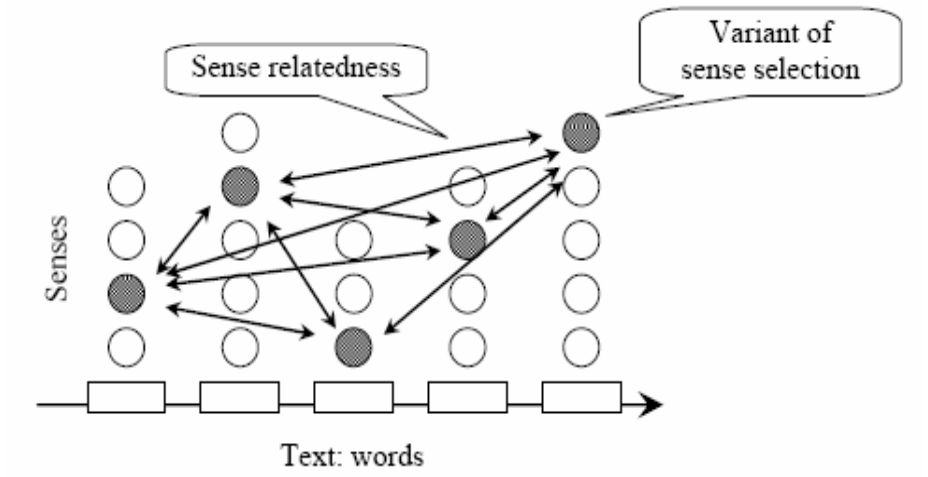

Figure 1. Graphic Representation of the Lesk Algorithm

For example: In performing disambiguation for the "pine cone" phrasal, according to the Oxford Advanced Learner's Dictionary, the word "pine" has two senses:

- sense 1: kind of evergreen tree with needle–shaped leaves,
- sense 2: waste away through sorrow or illness.

The word "cone" has three senses:

- sense 1: solid body which narrows to a point,
- sense 2: something of this shape whether solid or hollow,
- sense 3: fruit of a certain evergreen tree.

By comparing each of the two gloss senses of the word "pine" with each of the three senses of the word "cone", it is found that the words "evergreen tree" occurs in one sense in each of the two words. So these two senses are then declared to be the most appropriate senses when the words "pine" and "cone" are used together.

Acording to Padwardhan et al. [14] there are two hypotheses that underly this approach. The first is about the closeness of the words, i.e., the words that appear together in a sentence can be disambiguated by assigning to them the senses that are most closely related to their neighboring words. The idea behind this hypothesis is, the words that appear together in a sentence generally are related in some way, because to express some idea the human needs a set of words working together. The second hypothesis is that related senses can be identified by finding overlapping words in their definitions. The idea behind this is equally reasonable, in that words that are related will often be defined using the same words, and in fact may refer to each other in their definitions.

The major limitation to this algorithm is that dictionary glosses are often quite brief, and may not include sufficient vocabulary to identify related senses.

4 WordNet

WordNet is a large lexical database of English developed at Princeton University under the direction of George A. Miller. This resource combines several information used for word sense disambiguation. WordNet includes senses definitions as a dictionary, defines synsets and provide similarity relatedness between words.

WordNet is also freely and publicly available for download. WordNet's structure makes it a useful tool for computational linguistics and natural language processing.

We use WordNet 2.1 which contains 155327 instances, 17597 synsets and 207016 senses; is comprised by three databases of nouns and verbs and another for adjectives and adverbs.

5 Measures of Word Semantic Similarity

In order to quantify the degree to which two words are semantically related using information drawn from semantic networks, there are a number of measures that were developed, see [4] for an overview.

In selecting measures to analyze and compare, we focused on those that has better performance on word sense disambiguation. As a result, the four measures described below were selected. All these measures assume as input a pair of concepts, and return a value indicating their semantic relatedness.

We conduct our evaluation using the following similarity metrics: Jiang and Conrath, Lin, Lesk and a combination of measures [18]. We use the WordNet-based implementation of these metrics, as available in the WordNet::Similarity package [14]. We provide below a description of each of these four metrics.

5.1 The Jiang–Conrath Measure

This measure is based on another similarity measure given by Resnik [16], defined as follows. Resnik defines the notion of information content, wich is a measure of the specificity of a given concept, and is defined based on its probability of occurrence in a large corpus (1).

$$IC(C) = -\log(P(C)) \tag{1}$$

Given a text corpus, $P(C)$ is the probability of encountering an instance of type C. The value for $P(C)$ is therefore larger for concepts listed higher in the semantic hierarchy, and reaches its maximum value for the topmost concept (if the hierarchy has only one top, then the P value for this concept is 1). Startin with this concept of information content, Resnik defines a measure of semantic relatedness between words (2) by quantifying the information content of the lowest common subsumer (LCS) of two concepts (that is, the first common node in the semantic network encountered by traveling from the wo given concepts toward the root).

$$Similarity(C_1, C_2) = IC(LCS(C_1, C_2)) \tag{2}$$

Jiang and Conrath's [8] alternative to Resnik's definition (3) uses the difference in the information content of the two concepts to indicate their similarity (3).

$$Similarity(C_1, C_2) = 2 \times IC(LCS(C_1, C_2)) - (IC(C_1) + IC(C_2)) \tag{3}$$

5.2 The Lin Measure

The Lin [11] measure of semantic relatedness of concepts is based on his Similarity Theorem. It states that the similarity of two concepts is measured by the ratio of the amount of information needed to state the commonality of the two concepts to the amount of information needed to describe them. The commonality of two concepts is captured by the information content of their lowest common subsumer and the information content of the two concepts themselves. This measure turns out to be a close cousin of the Jiang–Conrath measure, although they were developed independently:

$$Similarity(C_1, C_2) = \frac{2 \times IC(LCS(C_1, C_2))}{IC(C_1) + IC(C_2)} \tag{4}$$

5.3 The Lesk Measure

As a solution for word sense disambiguation, Lesk [10] proposes to measure the relatedness between two concepts by the overlap between the corresponding definitions of them, as provided by a dictionary. The application of the Lesk similarity measure is not limited to semantic networks, and it can be used in conjunction with any dictionary that provides word definitions.

5.4 Combination of Similarity Measures

Sinha and Mihalcea [18] propose to implement a combination of the similarity measures, which accounts for the strength of each individual metric. They build a graph where use the similarity metric *jcn* to draw similarity values between *nouns* and the similarity metric *lch* to draw similarity values between *verbs*. All the other edges in the graph, including links between *adjectives* and *adverbs*, or links across different *parts-of-speech*, are drawn using the *lesk* measure.

6 Applying Similarity Measures on Original Lesk Algorithm

In order to compare and evaluate the different similarity measures described in Section 4, we implement the original Lesk algorithm changing the similarity measure proposed by Lesk for each of the similarity measures mentioned before. The senses chosen by the implementation are mapped to WordNet senses.

As we mention in Section 2, comparison for word sense disambiguation using similarity measures have been proposed by some authors [18][14], but none of these have used the original Lesk algorithm.

Moreover, there are other differences between our proposed implementation and the previous proposals, like:

- We evaluate the implementation on four different corpus: SENSEVAL-2, SENSEVAL-3, SEMEVAL and SEMCOR, while [18] used two and [14] used one.
- We do not make use of a predefined *window of context* as [14] that uses a window of three words, while our *window of context* is defined by the sentences size.

Also, in our experiments we consider two *back-off* strategies for words not covered by the implementation, consisting of the "most frequent sense" defined in WordNet and "random sense". This means that words for which all their possible meanings lead to zero relatedness with other word definitions are by default assigned sense number one in WordNet (most frequent sense) or random sense, depending of the strategy.

7 Experimental Results

7.1 Experimental Data

We carried out word sense disambiguation experiments using SENSEVAL-2 [12], SENSEVAL-3 [19], SEMEVAL [15] English all-words data sets and SEMCOR corpus.

Due to the complexity in time of the original Lesk algorithm, from some experimental data we eliminated sentenses with more than 210,567,168,000 combinations.

7.1.1 SENSEVAL

There are now many programs for automatically determining which sense of a given word is used in a text. One would like to be able to tell which is better, which worse, and also which words, or varieties of language, presented particular problems to which programs. Senseval is designed to meet this need.

The Senseval competition [20] is the first open, community-based evaluation exercise for word sense disambiguation. Started in 1997 following a workshop Tagging Text with Lexical Semantics: Why, What and How? [13], it is run by a small elected committee under the auspices of ACL-SIGLEX (the Association for Computational Linguistics' Special Interest Group on the Lexicon). It uses a DARPA-style evaluation format where the participants are provided with hand-annotated training data and test data and a pre-defined metric for evaluation. Unlike true DARPA evaluations, Senseval is a more grassroots exercise, self-initiated bye the WSD researchers. More than just a bake-off of automatic WSD systems, its underlying goal is to further to understanding of lexical semantics and polysemy [1].

Senseval has now had four competitions. In some of them they evaluated different tasks, like "English all-words". We use the annotated English text for this task provided by Senseval-2, Senseval-3 and Semeval (without the sentences more than 210,567,168,000 combinations, except Semeval).

SENSEVAL-2*:* This sample of the corpus consisted of 3 files with 2087 words (220 sentences), where 387 are adjetives, 252 adverbs, 992 nouns and 456 verbs.

SENSEVAL-3*:* This sample of the reduced corpus consisted of 3 files with 1563 words (238 sentences), where 238 are adjetives, 12 adverbs, 716 nouns and 597 verbs.

SEMEVAL: This corpus consisted of 3 files with 476 words (101 sentences), where 163 are nouns and 313 verbs.

7.1.2 SEMCOR

Semcor is a textual corpus (created at Princeton University) in which words are syntactically and semantically tagged. The texts included in Semcor were extracted from the Brown corpus and then linked to senses in the WordNet lexicon. All the words in the corpus have been syntactically tagged using Brill's part of speech tagger; the semantically tagging was done manually for all the nouns, verbs, adjectives and adverbs, each of these words being associated with its correspondent WordNet sense.

We are using the version 2.1 with a total of 352 files. In this corpus we take a sample consisted of 21 files with 12300 words (2053 sentences), where 1484 are adjetives, 877 adverbs, 5082 nouns and 4857 verbs.

7.2 Results

We use the original Lesk Algorithm with a window of context defined by the sentence size. Word sense disambiguation was evaluated based on the original Lesk algorithm using each of the similarity measures discussed above.

Results are reported for each corpus separately. We present the overall accuracy, where the number of correct instances is divided by the total number of instances. The results for the Jiang–Conrath, Lesk and combination of similarity measures are

shown in the columns *jcn*, *lesk* and *comb*, respectively. Also, we reported the results for each *back-off* strategy.

7.2.1 Evaluations on SENSEVAL-2

With *back-off* to the most frequent sense:

Table 1. Senseval-2 accuracy, back-off most frequent sense

	Similarity Measures		
Size (words)	***jcn*** (%)	***lesk*** (%)	***comb*** (%)
2087	60.5	54.7	56.2

With *back-off* to a random sense:

Table 2. Senseval-2 accuracy, back-off random sense

	Similarity Measures		
Size (words)	***jcn*** (%)	***lesk*** (%)	***comb*** (%)
2087	50.4	54.6	55

7.2.2 Evaluations on SENSEVAL-3

With *back-off* to the most frequent sense:

Table 3. Senseval-3 accuracy, back-off most frequent sense

	Similarity Measures		
Size (words)	***jcn*** (%)	***lesk*** (%)	***comb*** (%)
1563	53	49.8	53.7

With *back-off* to a random sense:

Table 4. Senseval-3 accuracy, back-off random sense

	Similarity Measures		
Size (words)	***jcn*** (%)	***lesk*** (%)	***comb*** (%)
1563	45.2	49.6	48.4

7.2.3 Evaluations on SEMEVAL

With *back-off* to the most frequent sense:

Table 5. Semeval accuracy, back-off most frequent sense

	Similarity Measures		
Size	***jcn***	***lesk***	***comb***
(words)	(%)	(%)	(%)
476	35.3	36.6	37.4

With *back-off* to a random sense:

Table 6. Semeval accuracy, back-off random sense

	Similarity Measures		
Size	***jcn***	***lesk***	***comb***
(words)	(%)	(%)	(%)
476	32.6	36.3	36.3

7.2.4 Evaluations on Semcor

With *back-off* to the most frequent sense:

Table 7. Semcor accuracy, back-off most frequent sense

	Similarity Measures		
Size	***jcn***	***lesk***	***comb***
(words)	(%)	(%)	(%)
12300	63.9	55.5	61.6

With *back-off* to a random sense:

Table 8. Semcor accuracy, back-off random sense

	Similarity Measures		
Size	***jcn***	***lesk***	***comb***
(words)	(%)	(%)	(%)
12300	53.1	55.3	48.4

We report the accuracy results for the similarity measures described in Section 5 for each experimental data in tables above. In addition, we provide the results of the best similarity measure for each corpus and for different back-off strategy (most frequent sense and random sense, represented as MFS and RS respectively) in Table 9.

Table 9. Best similarity measure for each corpus

	Corpus			
back-off	***Senseval-2***	***Senseval-3***	***Semeval***	***Semcor***
MFS	Jcn	comb	Comb	Jcn
RS	Comb	lesk	lesk / comb	Lesk

In order to have a better understanding of the recall results for each similarity measures used for WSD with the Lesk algorithm, we present the scores for each corpus using most frequent sense as a system by itself, i.e., for all words in the text the system assigned sense number one in WordNet (most frequent sense).

Table 10. Accuracy of most frequent sense system

Corpus			
Senseval-2 **(%)**	***Senseval-3*** **(%)**	***Semeval*** **(%)**	***Semcor*** **(%)**
64	61.3	47.5	73.7

8 Analysis and Discussion

The experimental results show that the combination of similarity measures is more accurate than each measure separately. However, the Lesk measure shows better performance when the back-off is random sense and the Jiang–Conrath measure shows good results only when we use back-off most frequent sense. Also this measure decreases its accuracy around 10% when the back-off is random sense. With this information we can suppose that the Jiang–Conrath measure does not have good recall and it bases its decision on the back-off of the system. Otherwise, the results reported by the Lesk measure do not change, independently of the back-off strategy used, therefore the Lesk measures has a good recall and does not need use the information given for the back-off strategy. In order to understand better this point, the Table 10 shows the accuracy for most frequent sense system, i.e., for all senses in the text the systems gives sense number one in WordNet. With this information we can observe that the use of most frequent sense as a back-off strategy is not a good option due to this method increases the accuracy of the methods that have bad recall.

On the other hand, the accuracy of the similarity measures with different back-off strategy is between 45 and 60 percent in three of the corpus while in Semeval data set the accuracy is between 35 and 37 percent. This is probably due to this corpus has only nouns and verbs and the size of the sentence is at most 16 words.

9 Conclusions

In this paper we compared different similarity measures used for word sense disambiguation based on the original Lesk algorithm. As far as we know, no attempt has been made in the past to evaluate the original Lesk algorithm using similarity measures due to its computational complexity.

Through experiments performed on four corpora we have shown that the combination of similarity measures proves to be the most accurate for word sense disambiguation based on the original Lesk algorithm.

Also, we showed the results for WSD based on the Lesk algorithm using different similarity measures with two back-off strategies: most frequent sense and random sense; and we present an analysis of the behavior of each similarity measure for the back-off strategies.

Aknowledgements. This work was done under support of Mexican Government trough CONACyT (Project Number: 50206-H) and SIP-IPN (Project Number: 20091587), as well as SNI to the second author.

References

1. E. Agirre and Edmonds P., Eds. *Word Sense Disambiguation: Algorithms and applications*. Text, Speech and Language Technology Series, Springer, 2007, Vol. 33, ISBN: 978-1-4020-6870-6.
2. E. Altintas, E. Karsligil, and V. Coskun. A New Semantic Similarity Measure Evaluated In Word Sense Disambiguation, In *Proceedings of the 15th NODALIDA conference,* S. Werner (ed.), 2006, pp. 8–11 ISBN 952-458-771-8, ISSN 1796-1114.
3. S. Banerjee and T. Pedersen. An adapted Lesk algorithm for word sense disambiguation using WordNet. In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics*, CICLing 2002, Mexico City, February 2002.
4. Budanitsky and G. Hirst. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and Other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics*, Pittsburgh, June 2001.
5. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.
6. Gelbukh, G. Sidorov, and S. Yong. Evolutionary Approach to Natural Language Word Sense Disambiguation through Global Coherence Optimization. *Transactions on Communications*, 1(2), 2003, pp. 11-19.
7. G. Hirst and D. St-Onge, Lexical chains as representations of context for the detection and correction of malapropisms. In: C. Fellbaum (Ed.), *WordNet: An electronic lexical database*, Cambridge, MA: The MIT Press, 1998, 305–332.
8. J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*, Taiwan, 1997.

9. Leacock and M. Chodorow. Combining local context and WordNet sense similarity for word sense identification. In *WordNet, An Electronic Lexical Database*. The MIT Press, 1998.
10. M. Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone. In *Proceedings of SIGDOC '86*, 1986.
11. Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, Madison, WI, 1998.
12. M. Palmer, C. Fellbaum, S. Cotton, L. Delfs, and H. Dang. English tasks: all-words and verb lexical sample. In *Proceedings of ACL/SIGLEX Senseval-2*, Toulouse, France, 2001.
13. M. Palmer and M. Light. Introduction to the special issue on semantic tagging. *Natural Language Engineering*, 5(2): i-iv.
14. S. Patwardhan, S. Banerjee, and T. Pedersen. Using Measures of Semantic Relatedness for Word Sense Disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 241–257, Mexico City, Mexico, February, 2003.
15. S. Pradhan, E. Loper, D. Dligach, and M. Palmer. SemEval-2007 task 17: English lexical sample, SRL and all words. In *Proceedings of the 4th international workshop on semantic evaluations (SemEval-2007),* pp. 87–92, Prague, Czech Republic, 2007.
16. P. Resnik. Using information content to evaluate semantic similarity. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montreal, Canada, 1995.
17. Sebti and A. Barfroush. A new word sense similarity measure in wordnet. In *Proceedings of the IEEE International Multiconference on Computer Science and Information Technology*, October, 2008.
18. R. Sinha and R. Mihalcea. Unsupervised Graph-based Word Sense Disambiguation Using Measures of Word Semantic Similarity, In *Proceedings of the IEEE International Conference on Semantic Computing (ICSC 2007)*, Irvine, CA, September 2007.
19. B. Snyder and M. Palmer. The English all-words task. In *Proceedings of ACL/SIGLEX Senseval-3*, Barcelona, Spain, July 2004.
20. Senseval. *Evaluation Exercises for the Semantic Analysis of Text* http://www.senseval.org