A new semantic similarity measure evaluated in word sense disambiguation

Ergin Altintas Naval Science and Eng. Inst. Turkish Naval Academy PK 34492 Tuzla, Istanbul ealtintas@dho.edu.tr

Elif Karsligil Computer Eng. Dep. Yildiz Technical University PK 34349 Yildiz, Istanbul elif@ce.yildiz.edu.tr

Vedat Coskun Naval Science and Eng. Inst. Turkish Naval Academy PK 34492 Tuzla, Istanbul vedatcoskun@dho.edu.tr

Abstract

In this paper, a new conceptual hierarchy based semantic similarity measure is presented, and it is evaluated in word sense disambiguation using a well known algorithm which is called Maximum Relatedness Disambiguation. In this study, WordNet's conceptual hierarchy is utilized as the data source, but the methods presented are suitable to other resources.

1 Introduction

Semantic similarity is an important topic in natural language processing (NLP). It has also been subject to studies in Cognitive Science and Artificial Intelligence. Application areas of semantic similarity include word sense disambiguation (WSD), information retrieval, malapropism detection etc.

It is easy for humans to say if one word is more similar to a given word than another. For example, we can easily say that *cruiser* is more similar to *destroyer* than *spoon* is. In fact, semantic similarity is a kind of semantic relatedness defining a resemblance.

There are mainly two approaches to semantic similarity. 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 (Fellbaum, 1998). As in many other NLP studies, hybrid approaches that make benefit from both techniques also exist in semantic similarity.

There is not many ways to evaluate a semantic similarity measure. You may check the correlation between your results and human judgments, or else you may select an application area of semantic similarity, and you compare your similarity measure with others according to the success rates in that application area.

In this study we have chosen the second method as our evaluation method, and WSD as our application area to practice our similarity measure and compare our results with the others'.

WSD is one of the most critical and widely studied NLP tasks, which is used in order to increase the success rates of NLP applications like translation, information retrieval etc. WSD can be defined as the process of selecting the correct or intended sense of a word, occurring in a specific context. The set of candidate senses are generally available from a lexical database.

The main idea behind our evaluation approach is: The success rate of WSD should increase as the similarity measure's performance gets better.

The remainder of this paper is organized as follows: We discussed the related work in Section 2. Our similarity measure and the WSD algorithm that we have used in this study are described in Section 3. The performance of our measure is evaluated, and compared to others in Section 4. Some discussion topics are probed in Section 5, and the paper is concluded in Section 6.

2 Related work

To quantify the concept of similarity between words, some ideas have been put forth by researchers, most of which rely heavily on the knowledge available in lexical knowledge bases like WordNet. First studies in this area date back to Quilian's semantic memory model (Quilian, 1968) where the number of hops between nodes of concepts in the hierarchical network

S. Werner (ed.), *Proceedings of the 15th NODALIDA conference, Joensuu 2005*, Ling@JoY 1, 2006, pp. 8–11
ISBN 952-458-771-8, ISSN 1796-1114. http://ling.joensuu.fi/lingjoy/

specifies the similarity or difference of concepts.

Wu and Palmer's semantic similarity measure (WUP) was based on the path length between concepts located in a taxonomy (Wu and Palmer, 1994), which is defined as:

$sim_{wup}(c_1, c_2) = max\left(\frac{2*depth(lcs(c_1,c_2))}{len(c_1,c_2)+2*depth(lcs(c_1,c_2))}\right)$

Resnik introduced a new factor of relatedness (Resnik, 1995) called information content (IC), which is defined as:

$IC_{res}(c) = -logP(c)$

Similarity measures of Resnik (RES) (Resnik, 1995), Jiang and Conrath (JCN) (Jiang and Conrath, 1997) and Lin (LIN) (Lin, 1998) all relies on the IC values assigned to the concepts in an *is-a hierarchy*, but their usage of IC has little differences:

$sim_{res}(c_1, c_2) = IC\left(lcs(c_1, c_2)\right)$

$rel_{jcn}(c_1, c_2) = \frac{1}{IC(c_1)+IC(c_2)-2*IC(lcs(c_1,c_2))}$

$rel_{LIN}(c_1, c_2) = \frac{2*IC(lcs(c_1,c_2))}{IC(c_1)+IC(c_2)}$

Using a different approach Hirst G. and St-Onge assigns relatedness scores to words instead of word senses. They set different weights for different kinds of links in a semantic network, and uses those weights for *edge counting* (Hirst and St-Onge, 1997).

The similarity measure of Leacock and Chodorow (LCH) is based on the shortest path length between two concepts in an *is-a hierarchy* (Leacock et al., 1998). The formulation is as follows:

$sim_{lch}(c_1, c_2) = max\left(-log\frac{ShortestLen(c_1,c_2)}{2*TaxonomyDepth}\right)$

3 Algorithms

3.1 Maximum Relatedness Disambiguation

In this study we have used a relatively simple algorithm named *Maximum Relatedness Disambiguation* which is sometimes also called as the *Adapted Lesk Algorithm* (Pedersen et al., 2003). This algorithm uses a quantitative measure of relatedness (hence similarity) between word senses as a measure to disambiguate them in a context.

In this algorithm, it is assumed that, the senses having higher similarity values with the senses of other words in the context, are more likely to be the intended sense. This assumption is the key of this algorithm to determine the intended sense of a target word occuring in a context.

1: Select a window of n-word size context that contains the target word in the middle.
2: Identify candidate senses of each word in the context.
3: **for** each candidate sense of the target word **do**
4: Measure the relatedness of the candidate sense of the target word to those of the surrounding words in the context.
5: Sum the relatedness scores for each combination of senses.
6: Assign this sum to the candidate sense of the target word.
7: **end for**
8: Select the candidate sense that has the highest score of relatedness.

In short, this algorithm assigns a target word, the sense that is most related (or similar) to the senses of its neighboring words. We have used this algorithm in order to evaluate our similarity measure, hence similarity is also a kind of relation.

3.2 Our similarity approach

We propose a new model based on the hierarchical structure of taxonomies, with which we tried to improve over LCH. We assume that WordNet's taxonomic structure is well organized in a meaningful way, so that the leaf nodes of the taxonomies are the most specific concepts in the hierarchy, and as we go up to the roots the specificity of concepts decreases. For noun taxonomies, one can also say that the root nodes are the most abstract ones, and as we go down to leafs, concreteness of the nodes increases.

We argue that, *concreteness* and *abstractness* are attributes of concepts which can help us improve our estimations when calculating similarity of concepts. Let's assume that we have three concepts c_1, c_2, c_3 and the shortest path length between c_1 and c_2 and the shortest path length between c_1 and c_3 is equal to 7.

In this case, according to the sim_{lch} formula, similarity of c_1 and c_2 is the same as similarity of c_1 and c_3. So, if we use sim_{lch} as the similarity measure for WSD, we will not be able to differentiate between c_2 and c_3. Let's assume that, c_1 is a leaf node and:

$Depth(c_1) = 5, ClusterDepth(c_1) = 5,$
$Depth(c_2) = 7, ClusterDepth(c_2) = 8,$
$Depth(c_3) = 4, ClusterDepth(c_3) = 8.$

By *ClusterDepth* we mean the depth of the

deepest node in a cluster. If we define the specificity of a concept using the hierarchical place in its local cluster as:

$Spec(c) = \frac{Depth(c)}{ClusterDepth(c)}$

Which will always be in the range $[0..1]$. Then we can calculate the specificity of these concepts as:

$Spec(c_1) = 5/5 = 1$

$Spec(c_2) = 7/8 = 0.875$

$Spec(c_3) = 4/8 = 0.5$

Then according to these specificity values, we may say that c_2 is nearly as specific as c_1 but c_3 is not. So, we can say that c_2 should be more similar to c_1, than c_3.

If we formularize our similarity measure, it has two components *LenFactor* and *SpecFactor* which are defined as:

$LenFactor = \frac{ShortestLen(c_1,c_2)}{2.TaxonomyDepth}$

$SpecFactor = abs(Spec(c_1) - Spec(c_2))$

and our similarity measure is defined as follows:

$sim_{our}(c_1, c_2) = \frac{1}{1+LenFactor+SpecFactor}$

If we assume *SpecFactor* to be zero in all cases, our measure behaves just like the LCH measure. So, we can easily say that the differentiator (from LCH) in our measure is the *SpecFactor*.

4 Evaluation

When a similarity measure for English is to be evaluated, it is usually compared to Miller & Charles' results (Miller and Charles, 1991) using correlation. Usually, the senses giving the maximum similarity score are considered to estimate a similarity score between two polysemous words, and this approach tends to give the best correlation values with Miller & Charles' results (Yang and Powers, 2005). But there is a possibility that the chosen senses (by the human judges) may not be the most similar ones, even though the estimated similarity score may be in correlation with the human judgments. In Miller & Charles' study the sense pairs chosen by the human judges were not explicitly stated. So, there is no easy way to discover if the senses selected by our algorithm are the same as the senses chosen by the human judges. Because of this, we didn't use this method of evaluation in our study.

Another way of evaluation is to analyze the similarity measures theoretically, but this may not be sufficient or practical for every case.

The approach which we have chosen, is to evaluate the similarity measures with respect to their performance within a particular NLP application (Budanitsky, 2001).

We did WSD experiments using the noun data of English lexical sample task of Senseval-2. Each instance was made up of three or four sentences containing a single tagged target word.

To access WordNet, we have used a Perl interface called WordNet::QueryData (Rennie, 2000), to compare our results with the existing measures, we have used the WordNet::Similarity package (Pedersen et al., 2004). We adopted our measure compatible to the WordNet::Similarity modules so that it can be published in the next release of the package.

Since taxonomies other than the noun taxonomy are very shallow in WordNet, we take only nouns as our target for disambiguation. We didn't PoS tag our input text, instead we used the approach in (Patwardhan et al., 2002), selecting the nearest words to our target word into our context, which have noun forms in WordNet, regardless of if they are used as a noun or not. Our results can be seen in Table 1.

Measure	Precision	Recall
RES	0.295	0.83
LCH	0.316	0.97
WUP	0.331	0.97
LIN	0.380	0.58
JCN	0.305	0.72
Our Measure	0.347	0.97

Table 1: Disambiguation results.

Our precision has %9.8 relative improvement over the LCH measure, %17.6 relative improvement over the RES measure, and %4.8 relative improvement over the WUP measure. Our recall is the same as the LCH and the WUP measures, which are also path-length based measures. The LIN measure has a better precision (relatively %9.5) than Our's, but the recall rate of the LIN measure is much lower (relatively %59.7).

5 Discussion

Although our precision rate is higher than the others, we think it is still smaller than what is needed. It seems that unnecessarily subtle distinction of senses in WordNet and the strict relation structure of WordNet are the cause for this. There are some techniques to overcome

this problem, which we plan to work on, in the future.

In our similarity approach all the leaf nodes have a specificity value of 1, but some leaf nodes in WordNet like *something* and *anything* are not as specific as their place in the hierarchy indicates. Although this kind of words are not too many, they can be identified manually, and filtered using a stop word list.

It should be noted that, a similarity measure may perform better than the other measures in a specific application area, but it may perform poor in some other application areas.

6 Conclusion

In this paper, we have introduced a new word sense similarity measure, and evaluated it in the WSD application area of NLP. We have only used a concept hierarchy. So, there is no sparse data problem in our approach.

All the source code and data used and developed in this study, can be downloaded from the author's web site[1].

References

Alexander Budanitsky. 2001. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures.

Christiane D. Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT Press, Cambridge, MA.

Graeme Hirst and David St-Onge. 1997. Lexical chains as representation of context for the detection and correction malapropisms.

Jay J. Jiang and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference on Research in Computational Linguistics*.

Claudia Leacock, Martin Chodorow, and George A. Miller. 1998. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, pages 265–283.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proc. 15th International Conf. on Machine Learning*, pages 296–304. Morgan Kaufmann, San Francisco, CA.

[1] Http://www.ergin.altintas.org

George A. Miller and Walter G. Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Process*, pages 1–28.

Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. 2002. Using semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*. Mexico City.

Ted Pedersen, Satanjeev Banerjee, and Siddharth Patwardhan. 2003. Maximizing semantic relatedness to perform word sense disambiguation. *Preprint submitted to Elsevier Science*.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. WordNet::Similarity - measuring relatedness of concepts. *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, pages 1024–1025.

M.Ross Quilian. 1968. Semantic memory. *Semantic Information Processing*, pages 216–270.

Jason Rennie. 2000. WordNet::QueryData: a Perl module for accessing the WordNet database. http://people.csail.mit.edu/∼jrennie/WordNet.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI*, pages 448–453.

Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexical selection. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 133 –138, New Mexico State University, Las Cruces, New Mexico.

Dongqiang Yang and David M. W. Powers. 2005. Measuring semantic similarity in the taxonomy of wordnet. In *ACSC*, pages 315–322.