A Divergence from Randomness Framework of WordNet Synsets' Distribution for Word Sense Disambiguation

Kostas Fragos[1], Christos Skourlas[2]

[1] Department Of Computer Engineering, NTUA,
Iroon Polytexneiou 9, 15780 Athens Greece

[2] Department Of Computer Science, TEIA,
Ag. Spyridonos 12210 Athens Greece

Abstract. We describe and experimentally evaluate a method for word sense disambiguation based on measuring the divergence from the randomness of the WordNet synsets' distribution in the context of a word that is to be disambiguated (target word). Firstly, for each word appearing in the context we collect its related synsets from WordNet using WordNet relations, and creating thus the bag of the related synsets for the context. Secondly, for each one of the senses of the target word we study the distribution of its related synsets in the context bag. Assigning a theoretical random process for these distributions and measuring the divergence from the random process we conclude the correct sense of the target word. The method was evaluated on English lexical sample data from the Senseval-2 word sense disambiguation competition, and exhibited remarkable performance compared to / better than most known WordNet relations based measures for word sense disambiguation. Moreover, the method is general and can conduct the disambiguation task assigning any random process for the distribution of the related synsets and using any measure to quantify the divergence from randomness.

1 Introduction

The main task of the Word Sense Disambiguation (WSD) could be defined as the assignment of a word to one or more senses by taking into account the context in which the word occurs. Such senses are usually defined as references to a dictionary like WordNet lexical database [1], or a word thesaurus especially constructed for the disambiguation task.

The first systems were based on hand-built rule sets and only ran over a small number of examples. However, using these reference works and small vocabularies as a source of word sense definition and information many algorithms were presented [2], [3], with the hope that they could run on much wider lexicons.

Nowadays, the availability of word sense repositories, such as WordNet which makes a great number of fine-grained word sense distinctions, increased the interest for the realization of more demanding WSD and generally NLP applications that can take advantage of these sense distinctions [4],[5],[6],[7]. Moreover, the fact that the

Fragos K. and Skourlas C. (2006).
A Divergence from Randomness Framework of WordNet Synsets' Distribution for Word Sense Disambiguation.
In *Proceedings of the 3rd International Workshop on Natural Language Understanding and Cognitive Science*, pages 71-80
DOI: 10.5220/0002499700710080

various senses are linked together by means of a number of semantic and lexical relations makes WordNet a valuable resource for formulation of knowledge representation networks, a very popular feature among the computational linguistics researchers.

Using definitions from the WordNet electronic lexical database, Mihalcea and Moldovan [8] collected information from Internet for automatic acquisition of sense tagged corpora. Fragos et al. [9] used the glosses of WordNet to collect sense related examples from Internet for an automated WSD task. The work of Banerjee and Pederson [10] proposed a new research view by adapting the original Lesk algorithm [2] for WSD to WordNet. According to their algorithm, a polysemous word can be disambiguated by selecting the sense that have a dictionary gloss sharing the largest number of words with the glosses of adjacent (neighboring) words. Pedersen et al. showed in [11][12] that WSD could be carried out using measures that are able to illustrate ("to score") the relatedness between senses of a word.

Apart from the use of (dictionary) definitions, much work has been done in WSD using the WordNet hyponymy/hypernymy relation. Resnik [5] disambiguated noun instances calculating the (semantic) similarity between two words and choosing the most informative "subsumer" (ancestor of both the words) from an IS-A hierarchy. In another approach Leacock and Chodorow [13] based on WordNet taxonomy proposed a measure of the semantic similarity by calculating the length of the path between the two nodes in the hierarchy. Agirre and Rigau [4] proposed a method based on the conceptual distance among the concepts in the hierarchy and provided a conceptual density formula for this purpose.

Both WordNet definitions and the hypernymy relation are used by Fragos et al. in [9], where the “Weighted Overlapping” Disambiguation method is presented and evaluated. The method extends the Lesk’s approach to disambiguate a specific word appearing in a context (usually a sentence). Senses’ definitions of the specific word, the “Hypernymy” relation, and definitions of the context features (words in the same sentence) are retrieved from the WordNet database and used as an input of their Disambiguation algorithm.

In this work we make a completely different hypothesis to evaluate the measures of relatedness between the context of the target word and its senses. Rather than looking for quantitative measures of relatedness we focus on qualitative features of relatedness. WordNet links each lexical entry (a set of synonyms called synset that represents a sense) with other lexical entries via semantic and lexical relations creating a set of related synsets. Using these relations, we can expand the context (the adjacent / surrounding words) of a word that is to be disambiguated. More precisely, the set (collection) of the related synsets of all the words in the context is used as a random sample and we study the (composite) distribution of the related synsets for each sense, and count the actual presences of the synsets in the sample. Then, we make the hypothesis that the related synsets are distributed randomly in the context sample, and we eventually assign a model of randomness in the distribution of the related synsets. Expecting that the correct sense will demonstrate a different behavior, as far as the distribution of its related synsets in the context set, than the others, we try to catch this differentiation by measuring the divergence from randomness and assign thus the correct sense to the target word.

The Kullback-Leibler divergence (KL-divergence), which is a measure of how different two probability distributions are, is used as the measure of divergence between the theoretical distribution, that is derived from the hypothesis about the model of randomness and the actual distribution observed in the data. The sense whose distribution has the least divergence from the model of randomness is selected as the correct sense for the target word. As far as the model of randomness, we assign to the related synsets and evaluate three alternative theoretical distributions: the standard Normal distribution, the Poisson distribution and the Binomial distribution. In the same framework, any model of randomness could be assigned to the data and any measure of differentiation between distributions could be used to quantify the "discrepancy" between the theoretical and the actual distribution.

In section 2, we describe the WordNet relations used by our algorithm to form the bags of the related synsets. In section 3, we describe our algorithm and how it works with the various models of randomness. In section 4, experimental results and a comparison with the results of other systems are given. Finally, some aspects of our method and future activities are discussed in section 5.

2 WordNet

WordNet is an electronic lexical database developed at Princeton University in 1991 by Miller et al. [1] and has become last years a valuable resource for identifying taxonomic and networked relationships among concepts.

Lexical entries in WordNet are organized around logical groupings called synsets. Each synset consists of a list of synonymous words, that is, words that could be interchangeable in the same context without variation in the meaning (of the context). Thus, the synset

{*administration, governance, establishment, brass, organization, organisation*}

represents the sense of governing body who administers something. The basic feature that differentiates WordNet from the other conventional dictionaries is the relations, pointers that describe the relationships between this synset and other ones. WordNet makes a distinction between semantic relations and lexical relations. Lexical relations hold between word forms; semantic relations hold between word meanings. Since a semantic relation is a relation between meanings, and since meanings can be represented by synsets, we must think of semantic relations as pointers between synsets. For each synset in WordNet, such pointers connect the synset with other ones and form a list of connected synsets (the "related synsets"). WordNet stores information about words that belong to four parts-of-speech: nouns, verbs, adjectives and adverbs. Prepositions, conjunctions and other functional words are not included. Besides single words, WordNet synsets also sometimes contain collocations (e.g. fountain pen, take in) which are made up of two or more words but are "treated" like single words. Our algorithm makes use of a portion of all the relations provided by WordNet for nouns, verbs, adjectives and adverbs, but we have also the possibility to use in a similar way any combination of these relations to achieve better results. We give a short description below for the relations used in our work.

In the case of nouns and verbs, the "*hypernymy / hyponymy*" and the "*antonymy*" relations are used (by our disambiguation algorithm) to form the bags of the related synsets. Based on some preliminary experimentation we did not work with all the possible combinations of WordNet relations, and we eventually concluded that the particular combination of these three WordNet relations results in a better disambiguation performance. In the case of adjectives, the *antonymy* and *similar to* relations are used by our algorithm since hypernymy/hyponymy is not available for adjectives. These relations are briefly described in this section.

Definitions of common nouns typically consists of "a superordinate term plus distinguishing features" [1]; such information can provide the basis for organizing nouns in WordNet. Hence, nouns are organized into hierarchies based on the "hypernymy/hyponymy", or "is-a", or "is a kind of" relation between synsets. For example, if the "is-a" relation is represented as => then we can form a tree hierarchy for the synset {*aid, assistant, help*} following the superordinate terms as they are defined in WordNet:

{*aid, assistant, help*} => {*resource*} => {*asset, plus*} => {*quality*} => {*attribute*} => {*abstraction*}

"*Hyponymy*" and "*hypernymy*" relations are used between nouns. They are also used between verbs with a slightly different manner. The examination of the hyponyms of a verb and their superordinates terms shows that lexicalization involves many types of semantic elaborations across different semantic fields [1]. These elaborations have been merged into a relation called "*troponymy*" (from the Greek word tropos that means, way, manner or fashion). This relation between verbs can be expressed using this way: verb synset V_1 is hypernym of V_2 if V_2 is into V_1 in some particular manner. V_1 is then the troponymy of V_2.

"*Antonymy*" is a lexical relation that links together two words that are opposites in meaning. It is used both for nouns and verbs in a similar way.

The "*antonymy*" is the most frequent relation for the adjectives in WordNet. Adjectives are arranged into clusters containing the head synsets and the satellite synsets. Each cluster is organized around these antonymous pairs. These pairs are indicated in the head synsets of a cluster. The majority of the head synsets have one or more satellite synsets, the role of which is to represent a concept that is similar in meaning to the concept represented by the head synset. The "*similar to*" is another frequent relation defined for adjectives. This is a semantic relation that links synsets of two adjectives that are similar in meaning, but are not enough close to be stored into the same synset.

3 The Divergence from Randomness Framework for Word Sense Disambiguation

The main task of a disambiguation system is to determine which of the senses of an ambiguous word (target word) must be assigned to the word within a linguistic context. Each word has a finite number of discrete senses stored in a sense inventory (the WordNet in our case) and the disambiguation algorithm, based on the context, must select among these senses the most appropriate for the target word.

3.1 Bags of Related Synsets

An important factor that influences the performance of the disambiguation algorithm is the appropriate use of the linguistic information derived from the context in which the target word is appearing. Local information provides valuable information for word sense identification. Leacock and Chodorov [13] experimented with a local context classifier and used windows specifying adjacent words around the target word in the (local) context. They concluded that an optimal value for the size of the window of the local context is ±6 *opened-class* words around the target one. *Opened-class* words are the words that are tagged as nouns, verbs, adjectives and adverbs by the part-of-speech tagger. Local information can provide a strong indication for the correct sense of the target word when its senses are not related each other. In this case, a large window would be very effective for identifying senses. Since local contextual clues occur throughout a text, statistical approaches that use the local context fill in the sparse training space by increasing the size of the context window. Gale et al. [14] found that their Bayesian classifier works most effectively with a window of ±50 words around the target one.

In our algorithm a different approach is used. Instead of counting words around the target one and specifying the best context window it seems better to work with a set of sentences of the context. This set is consisted of the sentence that contains the target word and one to three surrounding sentences. That is the format of the context for a target word in the Senseval-2 English lexical sample data over which we evaluated our algorithm.

To create the set of related synsets for the context we do not use any part-of-speech tagging procedure to tag the words. Hence, for all senses of each word in the context including the target one and for each part-of-speech category (nouns, verbs, adjectives and adverbs), we look up WordNet to find related synsets using the *antonymy*, *hypernymy* and *hyponymy* relations. To disambiguate a word we give the word itself and its part-of-speech (pos) category. Hence, for each sense of the target word and for the explicit pos category we look up WordNet and create separate sets of related synsets. In the case of disambiguating nouns and verbs we make use of the three WordNet relations *antonymy*, *hypernymy* and *hyponymy*, while in the case of disambiguating adjectives the *antonymy* and *similar to* relations are used.

We have formed a set of related synsets for the context and a separate set for each sense. In the next sub-section we describe how our algorithm works to assign the correct sense to the target word.

3.2 The Disambiguation Algorithm

The key idea of the disambiguation algorithm is to assign a theoretical distribution in the related synsets of each sense and then to measure the divergence of this theoretical distribution from the actual distribution observed in the context set using the KL-divergence metric. Initially, the bags of the related synsets for the context and the senses of the target word are created as exactly described in the previous section. In the next stage, for each sense, a measure of discrepancy of its related synsets distribution from the theoretical distribution is calculated using the KL-divergence. Finally,

the algorithm selects as the correct sense, the sense whose distribution has the minimum discrepancy. The following pseudo-code describes how the disambiguation algorithm works:

```
  procedure CreateContextBag
    for each word wi of the context
      for each part of speech (pos)of wi
        for each sense of wi
          for each legal relation
            select from WordNet the related synsets;
  end;

  procedure CreateSenseBag(Sk:sense; Pos: part of speech)
   for the sense Sk and the Pos part of speech category
    for each legal relation
      select from WordNet the related synsets;
  end;

 Begin
   CreateContextBag;
   for each sense Sk
     begin
       CreateSenseBag;
       calculate the empirical distribution of the Sense
                                    Bag in the ContextBag;
       calculate the theoretical distribution of the
              sense Bag from the pdf of the random model;
       find the distance between empirical and
                          theoretical distribution;
     end;
   select as correct sense the sense with the minimum
                                                 distance;
end.
```

In the above pseudo-code, with the term pdf we mean the probability density function of the model of randomness and with the term *legal relation* we mean the part of the WordNet relations used in this work to create the bags of the related synsets (see section 3.1).

The empirical distribution for each related synset is calculated by the formula:

$$P(x) = \frac{x}{S} \tag{1}$$

Where x is the frequency of the observation of the sense related synset in the context bag and S the total sum of the frequencies of all the observations in the context bag.

The theoretical distribution is estimated at each point x from the probability density function (pdf) of the model of randomness which has been assigned to the distribution of the related synsets. For example, if we make the hypothesis that the related synsets of each sense are distributed in the context bag following the standard Normal distribution, then we use equation 2 to compute the probabilities at each point x.

$$Q(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} \tag{2}$$

In addition, to evaluate some different models of randomness, besides standard Normal distribution, we also assign to the related synsets two other random distributional models: the Poisson model and the Binomial model of randomness.

For the Poisson distribution the pdf is:

$$Q(x) = \frac{\lambda^x}{x!} e^{-\lambda} \tag{3}$$

We set the value of λ (the mean value of the distribution) equal to the average value of all the frequencies of the observations in the context bag.

For the Binomial distribution the pdf is:

$$Q(x) = \binom{n}{x} p^x q^{(n-x)} \tag{4}$$

The result *Q(x)* is the probability of observing *x* successes in *n* independent trials, where the probability of success in any given trial is *p* (q=1-p). We set the value of the parameter *n* to the total sum of the frequencies of the observations in the context bag and the value of the parameter p to the reciprocal of the total synsets in the context bag (*p=1/k*, where *k* the number of synsets in the context bag).

The above three models are evaluated in three separate experiments. In each experiment we compare the model of randomness with the empirical distribution using the *relative entropy* or *Kulback-Leibler* (*KL*) *distance* between two distributions

$$D(p \,||\, q) = p(x) \log \frac{p(x)}{q(x)} \tag{5}$$

We can think about the relative entropy as the "distance" between two probability distributions: it gives us a measure of how closely two probability density functions are. One technical difficulty is that *D(p||q)* is not defined when *q(x)* =0 but *p(x)>0*. We could tackle this problem (as we did in the experiments) dividing by the quantity (*q(x)+1*) (instead of *q(x)*).

4 Experimental Results

We evaluate our algorithm on the lexical sample data of the Senseval-2 competition of word sense disambiguation systems [15]. This is an extensively large corpus of the English language that was sampled from BNC-2, the Penn Treebank (comprising components from the Wall Street journal, Brown and IBM manuals) and web pages. The dictionary used to provide the senses inventory is WordNet version 1.7.1. The

test data as well as the scores attained from a number of contesting systems are freely available from the web site of senseval-2 organization.

The English lexical sample data consists of two sets of data: the *training* set and the *test* set. All the items contained in these two sets are specific to one word class; noun, verb and adjective and all the corpus instances have been re-checked consistently and found to belong to the correct word class. This takes the burden of part-of-speech tagging from the word sense disambiguation procedure. Our algorithm is an unsupervised one in the sense that it does not need any training. Therefore, we utilize only the test set data. The test set consists of 73 tasks. Each task consists of many occurrences (instances) of text fragments (context) in which the target word appears in. Each such instance has been tagged carefully by human lexicographers and one or more appropriate senses from the WordNet sense inventory have been assigned to the instance. The duty of the sense disambiguation algorithms is to return these sense tags. Each instance consists of the occurrence of the sentence that contains the target word (the word that is to be disambiguated) and one to three surrounding sentences that provide the context of the target word.

A small number of instances for which a WordNet sense number is not provided by the key file were rejected from the testing data. We also rejected a small number of instances, when the target word was tagged by lexicographers with a sense that was not one of the senses of the word itself but it was the sense of one of the compound words that contained the target word. This task leads to a test set consisting of 1474 instances of 29 nouns, 1627 instances of 29 verbs and 759 instances of 15 adjectives.

To evaluate the success of an information retrieval system or / and a statistical natural language processing model we usually make use of the concepts of *precision* and *recall*. If the results that the system must correctly retrieve form a target set (of results) then: *precision* could be defined as a measure of the proportion of the selected items that the system got correctly and *recall* is defined as the proportion of the target results that the system retrieved. In our case, all the English lexical sample test data is the target collection. In Senseval-2 word sense disambiguation competition the F-measure was used that is a combination of precision and recall given by the following form:

$$F = \frac{1}{a(1/P) + (1-a)(1/R)} \tag{6}$$

Where P is the precision, R the recall and α is a weight / factor that determines the importance given to precision and recall. This form is simplified to $2PR/(P+R)$ when an equal weight ($\alpha=1/2$) is given both to precision and recall.

Table 1 shows the results obtained for each model of randomness when evaluating our system on the Senseval-2 English lexical sample test data for the three part-of speech categories. To form the bags of the related synsets we use *antonymy*, *hypernym*y and *hyponymy* relations in the case of disambiguating nouns and verbs and *antonymy* and *similar to* relations in the case of disambiguating adjectives.

Table 1. Evaluation results of our algorithm on Senseval-2 English lexical sample data using three different models of randomness: the standard normal, the Poisson and the Binomial model.

	EVALUATION RESULTS			
Model of Randomness	**Nouns**	**Verbs**	**Adjectives**	**Overall Results**
Standard Normal	0.315	0.175	0.318	0.257
Poisson	0.309	0.172	0.318	0.253
Binomial	0.309	0.175	0.291	0.249

These results show that the standard model of randomness attains an F-measure of 0.257 and it is our more effective model for the disambiguation task.

Our algorithm performs better than well-known measures of similarity and relatedness, which are based on WordNet information and were evaluated on the same test data in [12]. Although our algorithm uses only the WordNet synsets as its input, it performs comparably to the first systems in the Senseval-2 word sense disambiguation competition [15].

5 Discussion - Future Activities

In this work we presented and evaluated a novel method for word sense disambiguation. Using a part of the WordNet relations, bags of related synsets are formed for the context and the senses of the target word. The Kullback-Leibler (KL) divergence is used to quantify the discrepancy between the actual distribution of the senses related synsets, in the context bag, and the theoretical random model.

A suitable modeling of the distribution of words contained in the glosses is likely to be a good indicator for the sense they define. This will be an important consideration for future work, in which we will be able to examine different WordNet aspects such as synonyms and gloss words together, as well as to make a systematic assessment of the performance of all the possible combinations between WordNet relations.

References

1. Miller G., Beckwith R., Fellbaum C., Gross D., Miller K.: Introduction to WordNet: An On-line Lexical Database, Five Papers on WordNet, Princeton University (1993).
2. Lesk M.: Automatic sense disambiguation: How to tell a pine cone from an ice cream cone, in Proceedings of the 1986 SIGDOC Conference, Pages 24-26, New York. Association of Computing Machinery (1986).
3. Sussna, M.: Word sense disambiguation for free-test indexing using a massive semantic network. In Proceedings of the 2nd International Conference on Information and Knowledge Management. Arlington, Virginia, USA (1993).
4. Agirre E. and Rigau G.: Word Sense Disambiguation Using Conceptual Density. Proceedings of 16th International Conference on COLING. Copenhagen, (1996).

5. Resnik P.: WordNet and distributional analysis: A class-based approach to lexical discovery. Statistically-Based Natural-Language-Processing Techniques: Papers from AAAI (1992).
6. McCarthy D., Koeling R., Weeds J. and Carroll, J.: Finding predominant word senses in untagged text. In Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume, 279–286, (2004).
7. Patwardhan S.: Incorporating dictionary and corpus information into a context vector measure of semantic relatedness, Master's thesis, University of Minnesota, Duluth (2003).
8. Mihalcea R. and Moldovan D.: Automatic Acquisition of Sense tagged Corpora. American Association for Artificial Intelligence (1999).
9. Fragos K., Maistros I. and Skourlas C.: Using Wordnet Lexical Database and Internet to Disambiguate Word Senses, in Proceedings of 9th Panhellenic Conference in Informatics, Thessaloniki Greece, 20-22 Oct. (2003).
10. Banerjee S., Pedersen T.: An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet, in Proceedings of Third International Conference on Intelligent Text Processing and Computational Linguistics (CICLING-02), Mexico City, Mexico (2002).
11. Padwardhan S., Banerjee S., Pedersen T.: Using measures of semantic relatedness for word sense disambiguation. In proceedings of the Fourth International Conference on Intelligent text Processing and Computational Linguistics, Mexico City, (2003).
12. Pedersen T., Banerjee S., Padwardhan S.: Maximizing Semantic Relatedness to Perform Word Sense Disambiguation. Preprint submitted to Elsevier Science, 8 March (2005).
13. Leacock C., Chodorow M.: Combining Local Context and WordNet 5 Similarity for Word Sense Disambiguation. Wordnet: An Electronic Lexical Database, Christiane Fellbaum (1998).
14. Gale W., Church W. K., Yarowski D.: A Method for Disambiguating Word Senses in a Large Corpus, in Computers and Humanities 26, 1992
15. http://www.sle.sharp.co.uk/senseval2, 2002.