Word Sense Disambiguation Using WordNet Relations

Kostas Fragos[1], Yannis Maistros[1], Christos Skourlas[2]

[1] Department of Computer Engineering, National technical University of Athens
15780 Zogarfou Athens, Greece
Email: {kfragos,maistros}@ece.ntua.gr
[2] Department of Computer Science, Technical Educational Institute of Athens
12210 Aigaleo Athens, Greece
Email: cskourlas@teiath.gr

Abstract. In this paper, the "Weighted Overlapping" Disambiguation method is presented and evaluated. This method extends the Lesk's approach to disambiguate a specific word appearing in a context (usually a sentence). Sense's definitions of the specific word, "Synset" definitions, 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 our Disambiguation algorithm. More precisely, for each sense of the word a sense bag is formed using the WordNet definition and the definitions of all the "Hypernyms" associated with the nouns and verbs in the sense's definition. A similar technique is used, for all the context words and the definitions of the "Hypernyms" (associated with the context nouns and verbs), to form a context bag. Then, a technique of assigning weights to words is applied. The weight for every word is inversely proportional to the hierarchy depth in the WordNet taxonomy of the associated "synset". Eventually, the disambiguation of a word in a context is based on the calculation of the similarity between the words of the sense bags and the context bag. The proposed method is evaluated in disambiguating all the nouns for all the sentences in the Brown files.

1 Introduction

Lesk [10] used dictionary definitions to disambiguate a polysemous word appearing in a context. According to Lesk's method, each lexicon definition is represented as a bag of words occurring in the definition. For a specific word the definitions of its senses are found in the lexicon and then a (separate) bag of words for each sense is formed. For all the other words, in the same context with the polysemous word, their definitions are retrieved and another bag of all the words occurring in the definitions is formed. To disambiguate, Lesk simply counts the number of common words between the context bag and each sense bag. The sense with the maximum score (common words) is selected.

Cowie [5] proposed a similar method using definitions from the Logman's Dictionary of Contemporary English (LDOCE) and improved the results applying a procedure of simulated annealing.

Using definitions from the WordNet electronic lexical database [6], Mihalcea and Moldovan [11] collected information from Internet for automatic acquisition of sense tagged corpora. Montoyo and Palomar [12] presented a method for automatic disambiguation of nouns. They used the "Specification Marks" that are similar to semantic classes in WordNet taxonomy, and refined their results using definitions.

Apart from the use of (dictionary) definitions, much work has been done in word sense disambiguation using the WordNet hyponymy/hypernymy relation. Resnik [14] 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.

Other approaches used WordNet taxonomy. Lee et al. [9] and Leacock and Chodorow [8] proposed a measure of the semantic similarity by calculating the length of the path between the two nodes in the hierarchy. Agirre and Rigau [1] proposed a method based on the conceptual distance among the concepts in the hierarchy and provided a conceptual density formula for this purpose. Budanitsky and Graeme [4] presented experimental results comparing the above systems in a real-word spelling correction system.

Voorhess [16] is dealt with the problem of the lack of "containment" of clear divisions in the WordNet hierarchy and defined some categories. Sussna [15] used a disambiguation procedure based on the use of a semantic distance between topics in WordNet. A weighting scheme using WordNet relations was proposed. The synonymy relation gets a weight of zero value and hypernymy, hyponymy, holonymy and meronymy relations are assigned weights in the range [1, 2]. Antonymy arcs are assigned the value 2.5.

We applied the Lesk's method to disambiguate Semcor corpus but the results were rather poor. A performance of 36.14% was estimated. Hence, the information derived, in such a way, from WordNet definitions is rather insufficient for disambiguation tasks.

In this paper, we propose an improvement of the Lesk's method. We try to improve the performance of the disambiguation task by using additional definitions based on the "Hypernymy / Hyponymy" relation to enrich further the bags of words. Only the definitions related to the "hypernyms" of the nouns and verbs found in the context words and the senses' definitions were used. An experiment showed us that using hyponymy relation did not improve the performance. A procedure of assigning weights to the words within the bag is also used. This weight is inversely proportional to the hierarchy depth of the related synset definition. This technique improved the performance and the disambiguation task. An accuracy of 49,95% was estimated. Testing was based on the Semcor files.

In the sequel, in section 2 the WordNet taxonomy is briefly described, in section 3, the disambiguation procedure is presented. A short description of the WordNet glosses (definitions) and the process of extracting the words (features) enclosed in the bags are given. Then, the way of assigning weights into the bag words is described. Eventually, an algorithm for estimating the correct sense is given. It is based on the calculation of the similarity between the bags. In section 4, the experimental results are presented. In section 5, a short discussion of the proposed disambiguation method and comparisons with similar sense disambiguation systems are given. Some directions for future work are also given.

2 The WordNet Taxonomy

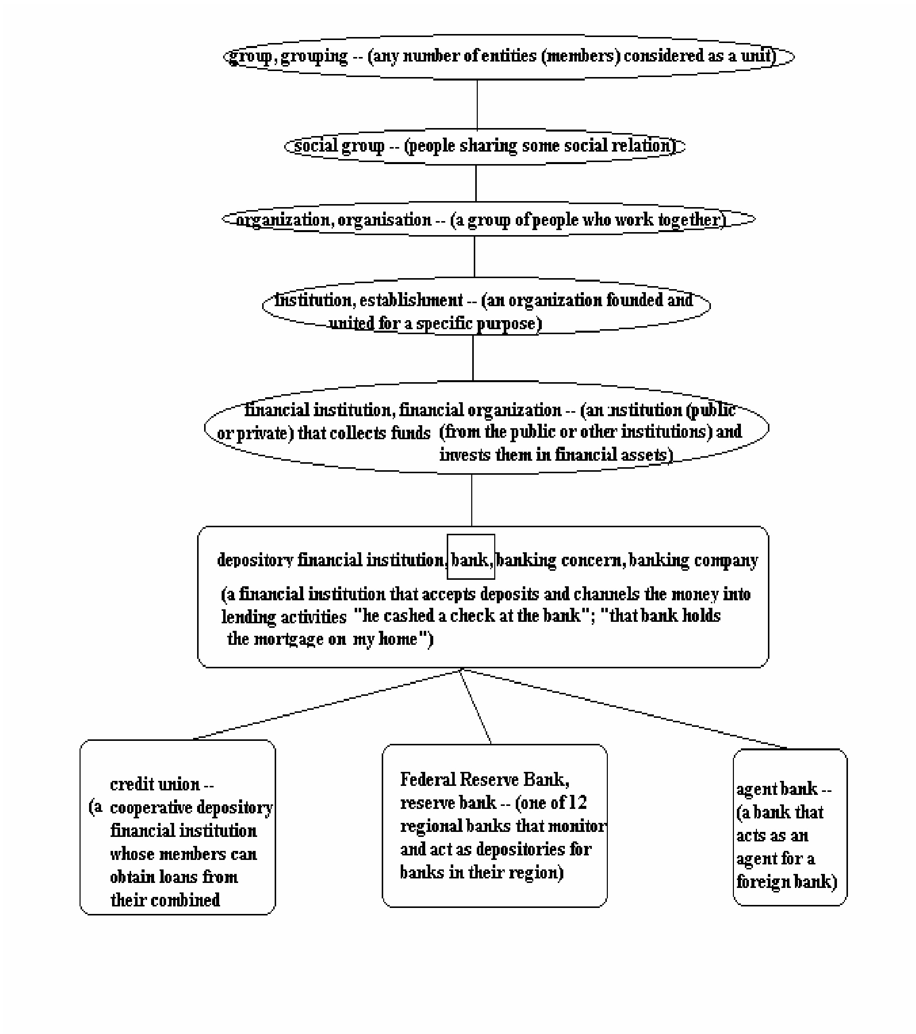

Fig. 1. Portion of the WordNet hypernymy/hyponymy relation for the word *bank* sense 1

The WordNet [6] is an electronic lexical database created at Princeton University in 1990. The WordNet organizes the lexical information in meanings (senses) and synsets (set of words – sentence(s) - describing the meaning of the word in a specific context). What makes WordNet remarkable is the existence of various relations between the word forms (e.g. lexical relations, like synonymy and antonymy) and the synsets (meaning to meaning or semantic relations e.g. hyponymy/hypernymy relation, meronymy relation). There are 31 such relations in WordNet vr. 6.1.

The hypernymy relation used in our method, is a part of the hyponymy/hypernymy relation. It organizes nouns and verbs into a lexical inheritance system. It is an IS-A hierarchy. A portion of WordNet hierarchy for the first sense of the word *bank* ("The financial institute") is shown in Figure 1. In this hierarchical system, a subordinate term inherits from the superordinate term the basic features and adds its own distinguishing features to form its meaning. Hence, the organization of the WordNet allows us to start from a topmost node or from an intermediate one and climb up or down finding the broader or narrower (more specific) meanings and then use them in a variety of ways. Figure 1 depicts, the synsets, a set of synomys, and the fact that each synset has a definition (gloss), which consists of a typical dictionary definition and some examples.
The entry for the word bank, with the sense of financial institution, is also shown in Figure 1. The four Sysnset words that are related to the meaning are the following:

Depository_financial_institution, bank, banking_ concern, banking_company

The defining phrase and the defining examples are enclosed into brackets separated by semicolon ';'.

3 The Disambiguation Procedure

As we have already mentioned, the proposed method extends the Lesk's approach for disambiguation, which is based on dictionary definitions. In this paper, the WordNet glosses are used.

3.1 The Way of Using WordNet Glosses

Initially, based on the defining part of a definition, a preprocessing phase takes place. It includes the part of speech tagging, tokenization and stemming. Then the features are extracted and enclosed within the bags. Unfortunately, the use of the defining examples of each synset definition (as an additional source of features) implied a reduction of the performance introducing some kind of "noise" information. Hence, for that reason, the defining examples are not included in the initial steps of the disambiguation procedure. Some words as articles, auxiliary verbs etc, namely, words conveying no significant information content, were rejected as useless in the first running of our algorithm using a stopwords list that removed them from the bags. But the adoption of removal those words showed that do not contribute to the disambiguation performance, so, we decided to include them in the bags and reject only those words with a number of characters less than three, such as a, or, in, of etc.

3.2 Features Preparation

All the definitions have to pass the preprocessing phase and then they could be used to find the hypernyms of nouns and verbs:
The part of speech tagging of the definitions is based on the Brill's tagger [Brill 92]. There was a need, for an implementation of the tagger, providing the possibility of a repeated invocation, during the execution, to parse the various WordNet definitions. Hence, an on-line version of the tagger was implemented in C++. As an example, the synset {*administration, disposal*} has the defining part "(*a method of tending to (especially business) matters*)" and the output is the following:

[*DT/a NN/method IN/of VBG/tending TO/to (VB/(especially NN/business) NNS/matters*]

Then, it is necessary to convert the words into the WordNet base forms (a task called "inflectional morphology") using a specific program developed for this purpose from the WordNet team.

The next phase is the *word stemming,* the process for removing the morphological and inflectional endings from the words before their use in various tasks e.g. information and text retrieval.

We decided to use stemming for two reasons: To keep the bag sizes and the processing time small for the evaluation experiment and a window open to future applications with the use of WordNet such us information and text retrieval etc. An implementation of the widely known Porter Stemming Algorithm [13] was used.

3.2 Assigning Weights

WordNet could be seen as the case of a semantic network representing knowledge in the form of interconnected nodes (the synsets) with edges (the relations). Various representational techniques related to semantic networks assign numerical values (weights) into edges. Such values depict different priorities in (or the importance of) traversing specific paths between the nodes e.g. Sussna [15].

Here, the proposed weighting scheme is simple. Every word within a bag is assigned a weight, depending on the depth of its related synset's position in the WordNet taxonomy (hierarchy). The assigned weight is inversely proportional to that synset's position.

Let us consider the disambiguation of a specific word. This word is appearing in a context with other words and it is also appearing in various WordNet synsets (its senses). In the beginning, a weight of 1 is assigned to all the synsets in which each context word appears and to all the synsets in all the senses of the word being disambiguated. Such synsets are called base synsets. We next parse the definitions of the base synsets and we assign the same weight of 1 to the extracted words. If some of these words are nouns or verbs, we scan the WordNet for their hypernyms and climbing up the hierarchy, in every level, we assign to the hypernymy synsets a weight inversely proportional to its distance from the base synset.

Let see the case of assigning weights to the synset {*administration, disposal*}. It is supposed that this synset is a base synset. Hence, it is found in the context or in a sense's definition of the word that is to be disambiguated. This synset has the defining part "(*a method of tending to (especially business) matters*)". Hence, the words *administration, disposal, method, tending, especially, business* and *matters*, after stemming, are entered into the corresponding bag and given a weight of 1. The nouns and verbs in the defining part are *method, tending, business* and *matters.* All belong to synsets and have their own hypernyms in WordNet taxonomy. *Method,* for example, belongs to two synsets (has two senses):

sense 1. *method - (a way of doing something, esp. a systematic one; implies an orderly logical arrangement (usually in steps))*
sense 2. wise, method - (a way of doing or being: "in no wise"; "in this wise")

Using the same way we extract the words (the synset words and the words in the defining parts) from the above definitions, and after stemming we assign a weight of value equal to 1 and put them again into the bag. The same process is repeated for the sense 2 and the synset {*wise, method}.*

Now, we completed the handling of the first level for the noun *method*, occuring in the base synset's definition, and go on with the hypernyms of the two senses. Here we examine the hypernym synsets only for the sense 1 (the synset {*method}).* All the hypernyms are listed below:

=> *know-how - (the (technical) knowledge and skill required to do something)*
=> *ability, power - (possession of the qualities (especially mental qualities) required to do something or get something done)*
=> *cognition, knowledge - (the psychological result of perception and learning and reasoning)*
=> *psychological_feature - (a feature of the mental life of a living organism)*

A weight of (1/2=0.5) is assigned to all the words extracted from the first hypernymy level {*know-how*}, to the synset words and the words from the defining part. Then we climb up a level in the hypernymy relation. Using the synset {*ability, power*} all the words are extracted and a weight of (1/3=0.3333) is assigned. Then a weight of (1/4= 0.25) is assigned to the words extracted from {*cognition, knowledge*} and a weight of (1/5=0.125) to the words extracted from the {*psychological_feature*}and so on.

3.4 The Algorithm

After the preparation phase described in the previous section 3.3 the features (words) are represented as bags of words related either with a sense or the context. A feature could only be inserted once into the same bag. Hence, we only count the features once. To disambiguate a word, two types of bags are used: A bag of words related to every sense of the word and a bag of words related to the context. The Lesk's approach [10] is based on the count of the common words between the bags related to each sense and the bag related to the context. The sense having the maximum overlapping with the context bag is chosen as the correct one. In our algorithm, this simple idea is used but the bags are extended with the use of features extracted from the definitions of the hypernyms. We also assign weights to the features. Figure 2 presents the basic components of the implemented disambiguation system. Part E of the Figure

2 depicts the calculation of the maximum overlapping between the sense's bags and the context bag.

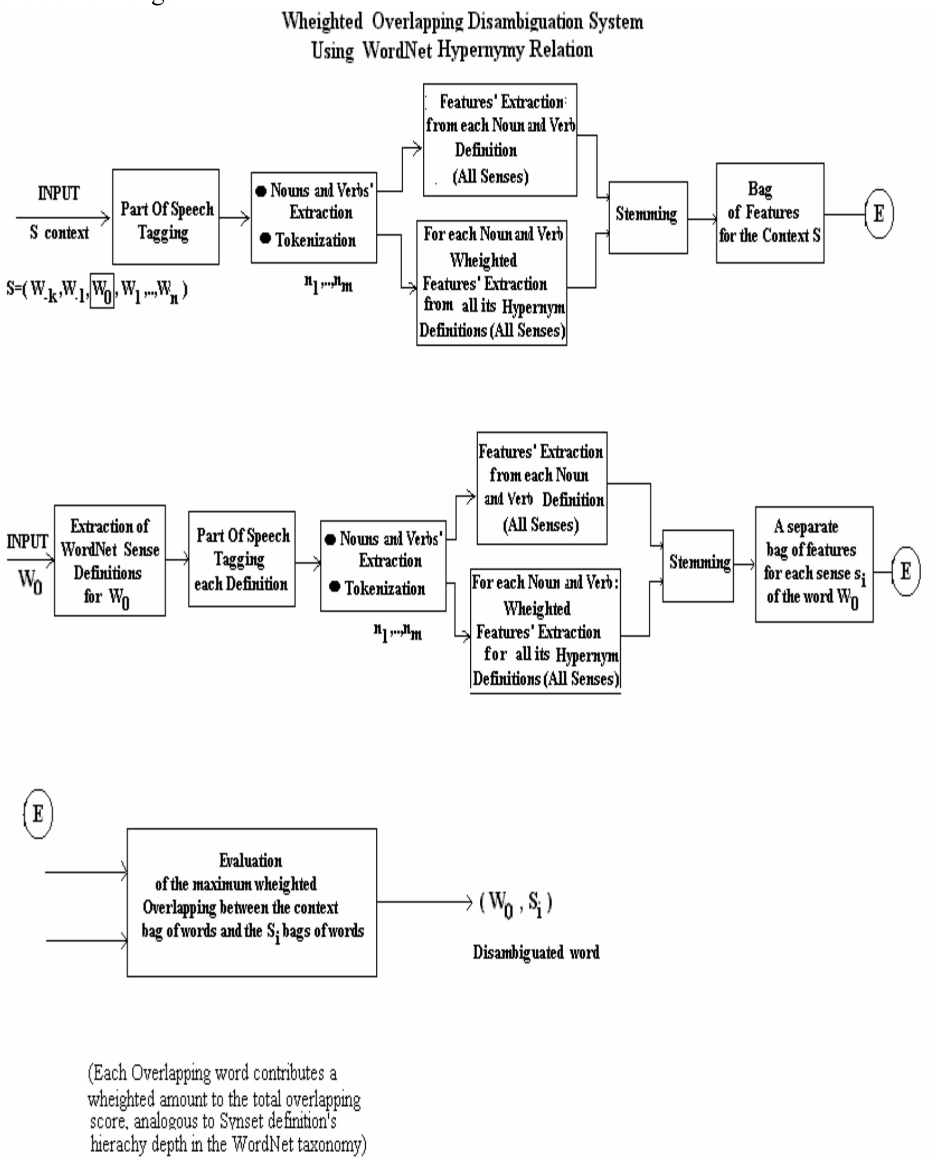

Fig. 2. The basic components of the disambiguation system

3.4.1 Maximum Overlapping Let us assume that (w_{-n}, w_{-n+1}, w_{-1}, w, w_1, w_2, ...,w_n) is the context of the word w that is going to be disambiguated. If B_c is the bag of the context words, and B_i, $i=1..k$ are the bags of the senses (of the word w) $s_1,s_2,\ldots,s_k$ and f_{jk} represents the j feature in the k definition then every bag could be seen as the union of such f_{jk} features: $B = \cup_{jk} f_{jk}$.

3.4.2 The "Weighted Overlapping" Disambiguation Algorithm The algorithm for the creation of the bags and the calculation of the maximum overlapping follows:

```
Procedure InsertIntoThebag(fj, B)
{
   If fj ∉ Bc Then
      Begin
         Assign the weight weigth(fj) to fj
         Bc ← fj;
      End;
}

Start: Read the context
For all wi i=-n To n, i<>0
   Begin
      Read its definition Dwi from WordNet
      For all fj ∈ Dwi InsertIntoThebag(fj,Bc);
      If wi is Noun or Verb
         For all hypernyms of wi
            Begin
               Read the definition Dh
               For all fj ∈ Dh InsertIntoThebag(fj,Bc)
            End;
   End;
For all Si i= to k
   Begin
      Read its definition Dsi from WordNet
      For all fj ∈ Dsi InsertIntoThebag(fj,Bi)
      If fj is Noun or Verb
         For all hypernyms of fj
            Begin
               Read the definition Dh
               For all fj ∈ Dh InsertIntoThebag(fj,Bi);
            End;
   End;

{Here is the calculation of the maximum overlapping}
For all senses si of w
   Begin
      Score(si) = 0;
      For each fj in Bi
         If fj = fk  in Bc then
         Score(si) = Score(si) + weigth(fj)* weigth(fk);
End

Choose as Correct Sense s s.t. s = arg maxsk score(sk);
```

4 Evaluation

Our disambiguation method was evaluated using the Semcor files [7]. The Semcor files are manually disambiguated text corpora using senses of WordNet vr. 1.6. They consist of 103 passages from the "Standard Corpus of Present-Day Edited American English" (the Brown Corpus) and the complete text of Stephen Crane's "*The Red Badge of Courage*".

Initially, we thought of using, as the context of the word *w,* a fixed word window around the *w.* Alternatively, we thought of testing the algorithm considering the context as a complete sentence (as it is exactly found within the Semcor text).

After some experimentation we decided to disambiguate all the noun appearances in Semcor files (75,000 noun occurrences in Brown 1 and Brown 2 corpus) using as context the complete sentence.

Table 1 depicts the comparison of our method with Lesk's method. We disambiguated Brown 1 and Brown 2 corpus. A portion of the disambiguation results from Brown 1 is listed in Table 2.

Semcor Files	Correct (%)	Ambiguous (%)
Lesk's Method	36.14	32,66
"Weighted Overlapping"	49,95	0,2

Table 1. Comparison with Lesk's method

Nouns	File		Correct		Correct (%)			%
Total=573	br-a01	Correct	309	573	53,93	Ambiguous	0/573	0
Total=611	br-a02	Correct	360	611	58,92	Ambiguous	0/611	0
Total=582	br-a11	Correct	316	582	54,3	Ambiguous	0/582	0
Total=570	br-a12	Correct	265	570	46,49	Ambiguous	0/570	0
Total=575	br-a13	Correct	326	575	56,7	Ambiguous	5/575	0,87
Total=542	br-a14	Correct	287	542	52,95	Ambiguous	0/542	0
Total=535	br-a15	Correct	269	535	50,28	Ambiguous	0/535	0
Total=505	br-b13	Correct	234	505	46,34	Ambiguous	0/505	0
Total=458	br-b20	Correct	242	458	52,84	Ambiguous	1/458	0,22
Total=512	br-c01	Correct	264	512	51,56	Ambiguous	1/512	0,2
Total=524	br-c02	Correct	256	524	48,85	Ambiguous	0/524	0
Total=452	br-c04	Correct	245	452	54,2	Ambiguous	6/452	1,33
Total=377	br-d02	Correct	168	377	44,56	Ambiguous	0/377	0
Total=466	br-d03	Correct	265	466	56,87	Ambiguous	0/466	0
Total=505	br-d04	Correct	217	505	42,97	Ambiguous	1/505	0,2
Total=446	br-e01	Correct	255	446	57,17	Ambiguous	0/446	0

Table 2. The results from the first 15 files of Brown 1 Corpus

As we can see from the two tables, the number of the ambiguous responses of our system is reduced dramatically in comparison with the responses of the Lesk's method. This is due to the increment of the bags words using the extra definitions from the hypernymy relation.

5 Discussion, Future Work and Comparison with Similar Works

In this paper we presented and evaluated a new disambiguation method, the "Weighted Overlapping method", based on the WordNet hypernymy relation only. The bags of words in the Lesk's disambiguation method are enhanced with additional words derived from the WordNet hypernymy relation. In each additional word, a weight inversely proportional to the hierarchy depth of its associated hypernym is given and a weighted contribution to the overlapping score is counted. This improves substantially the results of the Lesk's method. Unlike "Hypernymy" relation, the "Hyponymy" relation seems not to contribute to the disambiguation performance. An evaluation of the additional use of "Hyponymy" related definitions showed that it reduces the performance about 6%.

Apart from the taxonomic relations, several other semantic relations have been encoded in WordNet. Hence, it would be interesting to evaluate the "Weighted Overlapping" method, using additional definitions derived from these relations. We tried to use the coordinate relation. In WordNet, coordinate terms are the terms that have the same hypernymy synset and a great number of such terms are available for each synset in WordNet. A complete evaluation of this type of information was difficult to be done at the time being, because WordNet does not return for this relation linked lists of coordinate terms. In a future work, we will try to overcome this problem making a word sense disambiguation program that will exploit the available information from the coordinate relation. There is also a plan for experimentation with other semantic relations in WordNet.

The work of Banerjee and Pederson [2] presents an adaptation of the original Lesk Algorithm using WordNet semantic relations and definitions. In this work, the context of the ambiguous word is transformed into a set of combinations taking into account the various senses (for each context word). Such combinations are increased using senses from related synsets. Each combination is assigned a score by adding the overlaps between the definitions of the words belonging to this combination. The combination with the highest score is the preferred and the target word is assigned to the sense involved in this combination. This method presents some complexity as the number of possible combinations grows rapidly. This algorithm was evaluated on test data from the English lexical sample task used in SENSEVAL-2 (comparative evaluation of word sense disambiguation systems). In our method, a bag of words (approach) is used in a completely different way. We take nouns and verbs contained in a context word definition or in a sense definition and look up WordNet for their hypernyms definitions, in order to obtain additional terms and expand the vectors.

In another work [12], the glosses and the IS-A hierarchy of WordNet are used to form a disambiguation system for nouns. Using a series of heuristics, Montoyo and Palomar evaluate similarity between context and senses of the word. Then, they count the common words between context and glosses, taking all glosses coming from the definitions of the sense and the definitions of its hypernyms and hyponyms. This work resembles to our work only at the point of assigning weights to the common words. It takes into consideration their relative depth within the sub-hierarchy. The use of similarity measure in our work, the way of handling a WordNet definition and the use of hypernym definitions are different. The authors evaluated the method over a small part of Brown corpus and attained a performance of 52.5% with the base method and improved it with heuristics to 66.2%.

References

1. Agirre E., Rigau G.: Word Sense Disambiguation Using Conceptual Density. *Proceedings 16th International Conference on COLING*. Copenhagen (1996)
2. Banerjee S., Pedersen T.: An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. *Proceedings 3rd International Conference on Intelligent Text Processing and Computational Linguistics (CICLING),* Mexico City, Mexico (2002)
3. Brill E.: A Simple Rule Based Part of Speech Tagger. *Proceedings 3rd Conference on Applied Natural Language Processing (ACL),* (1992)
4. Budanitsky A., Graeme H.: Semantic distance in WordNet: an Experimental, Application-oriented Evaluation of Five Measures. *Proceedings Workshop WordNet and Other Lexical Resources.* The North American Chapter of the Association for Computation Linguistics (NAACL), Pittsburgh, PA, (2001)
5. Cowie J., Cuthrie J., Guthrie L.: Lexical Disambiguation using Simulated Annealing. *Proceedings DARPA Workshop on Speech and Natural Language,* New York (1992) 238-242
6. Fellbaum C.: WordNet, An Electronic Lexical Database. The MIT Press, Cambridge MA (1998)
7. Landes S., Leacock C., Tengi R.: Building Semantic Concordances. In: *WordNet, an Electronic Lexical Database,* MIT Press, Cambridge MA (1998) 199-216
8. Leacock C., Chodorow M.: Combining Local Context and WordNet5 Similarity for Word Sense Disambiguation. In: *Wordnet: an Electronic Lexical Database*. Christiane Fellbaum (1998)
9. Lee J.H., Kim H., Lee Y.J.: Information Retrieval Based on Conceptual Distance in IS-A Hierarchies. *Journal of Documentation*, 49(2):188-207 (1993)
10. Lesk M.: Automatic Sense Disambiguation: How to Tell a Pine Cone from an Ice Cream Cone. *Proceedings ACM SIGDOC Conference*, New York. (1986) 24-26
11. Mihalcea R., Moldovan D.: Automatic Acquisition of Sense Tagged Corpora. American Association for Artificial Intelligence (1999)
12. Montoyo A., Palomar M.: Specification Marks for Word Sense Disambiguation: New Development. *Proceedings CICLing Conference*, (2001) 182-191
13. Porter M.F.: An Algorithm for Suffix Stripping, In: Sparck Jones, Karen, and Peter Willet (eds.), *Readings in Information Retrieval*, Morgan Kaufmann, San Francisco (1997)
14. Resnik P.: Disambiguating Noun Groupings with Respect to WordNet Senses. *Proceedings 3rd Workshop on Very Large Corpora.* Cambridge, MA (1995) 54-68
15. Sussna M.: Word Sense Disambiguation for Free-text Indexing Using a Massive Semantic Network. *Proceedings 2nd International Conference on Information and Knowledge Management (CIKM).* Arlington, Virginia, (1993)
16. Voorhees, E.: Using WordNet to Disambiguate Word Senses for Text Retrieval. *Proceedings ACM SIGIR Conference*, (1993)