Using Wordnet Lexical Database and Internet to Disambiguate Word Senses.

Kostas Fragos
Dep. Of Computer Engineering
NTUA
Iroon Polytexneiou 9,
15780 Zogarfou
GREECE Athens
kfragos@ece.ntua.gr

Yannis Maistros
Dep. Of Computer Engineering
NTUA
Iroon Polytexneiou 9,
15780 Zogarfou
GREECE Athens
maistros@ece.ntua.gr

Christos Skourlas
Dep. Of Computer Science
TEIA
Ag Spyridonos
12210 Aigaleo
GREECE Athens
cskourlas@teiath.gr

Abstract.

The term "knowledge acquisition bottleneck" has been used in Word Sense Disambiguation Tasks (WSDTs) to illustrate / express the problem of the lack of large tagged corpora. In this paper, an automated WSDT is based on text corpora extracted / collected from Internet web pages. First, the disambiguation for the sense of a word, in a context, is based on the use of its definition and the definitions of its direct hyponyms in the WordNet to form queries for searching the Internet. Then, the "sense-related examples", in other words the collected answers / information, are used to disambiguate the word's sense in the context. A (similarity) metric is used to calculate the similarity between the context and the "sense-related examples" and the word is assigned the sense of the most similar example with the context. Some experiments are briefly described and the evaluation of the proposed method is discussed.

Keywords: Word Sense Disambiguation, Sense tagged Corpora, WordNet, Synset, Sense Definition, Hyponyms, Similarity metric, InterNet, Altavista, Searching Query.

1. Introduction

The word sense ambiguity is a hard problem for the developers of Natural Language Processing (NLP) systems. Words, often, have different meaning in various contexts. As an example, *bus* could mean a "vehicle" or an "electrical conductor connecting circuits". NLP related applications as text retrieval, automatic translation, summarization etc. are examples of the importance of word sense disambiguation.

Much work has been done to develop computer systems that receive plain text as input and tag each word following a disambiguation task. Two main approaches could be mentioned:

Knowledge based techniques, using Machine Readable Dictionaries (MRDs) and Thesaurus [Resnic 95], [Voorhess 93], [Sussna 93], [Yarowski 92], [Aggire & Rigau 96]. Corpus-based techniques usually using (disambiguated) text to train a statistical disambiguation module [Hearst 91], [Gale et al. 92], [Brown et al. 91], [Yarowski 93], [Leacock & et al.93].

Statistical approaches are considered good techniques but suffer from the problem of "data sparseness". In general, the larger the corpora the better the disambiguation accuracy. But the problem is that some words are infrequent even in the largest corpora. In other words, in order to disambiguate a polysemous word, this word must have a critical number of occurrences in a corpus. To overcome this problem, other techniques have been developed combining Knowledge-based and Corpus-based approaches:

In order to fill in the gaps in "sparse" training data, Resnik [Resnik 92] uses an information-based measure, "the most informative class", based on the WordNet taxonomy. Leacock and Chodorow [Leacock & Chodorow 98] in their experiment exploit similarity measures based on the distances between words in the WordNet taxonomy. They report a modest improvement in performance, especially when the training occurs on small data sets. Similarity-based techniques [Dagan et al. 94], [Carov & Edelman 97] calculate similarity measures between words based on co-occurrence in similar contexts. However, these approaches require also a substantial amount of training set. Luk [Luk 95] uses definitions from the Longman Dictionary of Contemporary English - LDOCE (Procter, 1978) and tries to find co-occurrence of concepts in a relatively small corpus, the Brown corpus, that consists of one million words. LDOCE is used because all its definitions have been written using a set of 2000 words. Luk reports that the proposed system achieves an average accuracy comparable to human performance given the same contextual information. The serious problem for this approach is that many words of the control vocabulary are polysemous. Thus, a defining concept actually stands for a number of different concepts.

Karov and Edelman [Carov & Edelman 97] proposed a similarity measure between words and sentences, with other sense-related examples ("feedback set"), to automate WSDT. Their system learns to disambiguate using as examples the appearances of a polysemous word in an untagged corpus. The "feedback set" for a sense is the union of all contexts that contain some noun found in the sense's definition (in a MRD). No mention is made of the way of collecting such data.

Mihalcea and Moldovan proposed a method [Mihalcea & Moldovan 99] for the automatic acquisition of sense tagged corpora. First, the WordNet is searched and the various senses of a word are determined. Then, the WordNet definition, for each possible sense, is used to form an Internet query that finds text examples containing only the exact defining phrase.

Finally in the collection of examples gathered from Internet the defining phrase is replaced with the original word form. Hence, example sentences for each sense are created. They report that the results were manually checked for correction and an accuracy of 92% was achieved based on human judgment.

Open Mind Word Expert in [Chlovski & Mihalcea 02] is an implemented active learning system that proposes a method for creating large sense tagged corpora that may be collected from the existing millions of web users. Mihalcea in [Mihalcea 02], also uses Internet, WordNet and Semcor corpus for the automatic generation of large sense tagged corpora.

In this paper, an adoption / modification of the similarity measure of Karov and Edelman is used. This similarity-based technique exhibits a good behavior, provided that a sufficient amount of tagged corpora is used during the training phase. The definitions for senses found in WordNet are used and the acquisition of sense-related examples ("the feedback set") is automated using information gathered from Internet with search engines. The WordNet's entries are organized into synonyms (sets) and various relations link the synonyms (sets). Hence, the hyponymy relation between nouns in WordNet taxonomy is used to enrich our data collection from search engines results.

In our method the Internet searches are used in a different way. For each sense, we do not search for exact phrases but we gather sentences containing only combinations of words in synsets and nouns found firstly in WordNet sense definition and secondly in WordNet definitions of all its direct hyponyms. These sentences ("the sense-related examples") disambiguate the word's examples using a similarity-based technique. In section 1 a brief introduction to WordNet lexicon is given and the sense entries and the semantic hierarchies are explained. In section 2 the semantic similarity measure, used in our method, is described. In section 3 the method is presented. Finally, in sections 4 and 5 experimental results are given and discussed.

1. An Introduction to the WordNet.

WordNet [Miller et al. 93], [Miller 93], [Fellbaum 98] is an electronic lexical database developed at Princeton University. WordNet entries ("senses") are organized into synonyms sets ("synsets") representing concepts. Each synset in WordNet is followed by its definition ("gloss") which contains a defining phrase, an optional comment and examples. WordNet supports two types of relations: semantic relations, which link concepts (i.e. synsets), such as hypernymy, hyponymy, meronymy, holonymy, troponymy etc. and lexical relations, such as antonymy, which links individual words. In this paper the hypernymy-hyponymy relation between noun senses is used. This relation generates a hierarchical semantic organization for nouns and verbs. It is expressed with bi-directional pointers between synsets. Table I depicts the synsets, the gloss, the hyponyms and the hypernyms of the sense *administration#2.*

administration#2 (#2 means the second sense of the word in Wordnet)
"*governing body who administers something*"
Synset : (*administration, governance, establishment, brass, organization, organisation*)

Gloss	
2. administration, governance, establishment, brass, organization, organisation -- (the persons (or committee or departments etc.) who make up a governing body and who administer something; "he claims that the present administration is corrupt"; "the governance of an association is responsible to its members"; "he quickly became recognized as a member of the establishment").	
Hyponyms	Hypernyms
=> executive -- (persons who administer the law) ⇨ judiciary, bench -- (persons who administer justice) ⇨ judiciary, bench -- (persons who administer justice) ⇨ management -- (those in charge of running a business)	⇨ body -- (a group of persons associated by some common tie or occupation and regarded as an entity; "the whole body filed out of the auditorium") => gathering, assemblage -- (a group of persons together in one place) => social group -- (people sharing some social relation) ⇨ group, grouping --(any number of entities (members) considered as a unit).

TABLE I: The synsets, the gloss, the hyponyms and the hypernyms of the sense *administration#2*

A measure of similarity among senses could be the conceptual distance among senses in Wordnet hierarchies [Miller & Teibel 91]. This distance is defined as the length of the shortest path between two senses in this hierarchical semantic net. However, Leacock and Chodorow [Leacock & Chodorow 98] pointed out that WordNet similarity measure is inadequate as a stand-alone classifier for word sense disambiguation.

2. A method for calculating (Semantic) Similarity.

In this paper, the similarity-based technique used by Carov and Edelman [Carov & Edelman 97] is modified and used. More precisely, the similarity between the word *w* in a context and the sense-related examples, for each sense of *w,* is calculated. Carov and Edelman reported that the use of weights (based on the total frequency of a word in the corpus or textual distance from the target word etc) contributed about 5% to the disambiguation performance. Hence, our calculation did not use weights to emphasize the importance of contribution between words etc. However, the definitions of all the direct hyponyms of a “target” word were additionally used. Hence, our sense-related examples consist of co-occurrences of nouns that are found both in the definition of the sense and the definitions of all its direct hyponyms.

Our technique, also, incorporates/uses two similarity measures: word similarity measure and context similarity measure. Words are considered similar if they are appeared to similar contexts and contexts are similar if they contain similar words. These definitions allow us to identify similarities for words and sentences using an iterative process.

<u>Example</u> (adopted from Carov and Edelman).

Let us consider the following sentences:

S1: *eat banana.*
S2: *taste banana.*
S3: *eat apple.*

In the first iteration, the word *banana* is considered to be similar with the word *apple* because they appear in the same contexts (notice verb *eat*). The words *eat* and *taste* are similar because they appear in similar contexts (notice the word banana).

In the second iteration, sentences S2 and S3 are considered to be similar because they share similar words (*taste, eat*) and (*banana, apple*) identified in the previous iteration. As a consequence of the similarity between sentences S2, S3, we can conclude the similarity between the words: *taste* and *apple* (they appear in similar sentences).

The example is focussed on two important aspects of this technique: the iteration in detecting similarities and the transitivity of such a consideration of similarity, which allows us to catch high-order similarities.

To formalize the intuitive definitions of similarity, we assume that each sentence is represented as a set of features: Nouns, verbs (not including auxiliaries), and adjectives.

2.1 Notation

The similarity between sentences S_1 and S_2, in n iteration, is denoted as $\mathrm{sim}_n(S_1,S_2)$ and the similarity between words W_1 and W_2, in n iteration, as $\mathrm{sim}_n(W_1,W_2)$. The fact that a word w belongs to a sentence s is denoted as $w\in s$ and if a sentence s contains a word w it is denoted as $s \ni w$.

The similarity between sentences S_1 and S_2 is defined in the following way:

$$\mathrm{sim}_{n+1}(s_1,s_2) = \Big(\sum_{w\in S_1} \max_{W_i\in S_2} \mathrm{sim}_n(w,w_i) \Big) / m \quad \text{(equation 1)}$$

where m is the number of the words contained in sentence S_1.

In other words, the similarity between sentences S_1 *and* S_2, in iteration n+1, for all $w\in S_1$, is the average of all the max $\mathrm{sim}_n(W,W_i)$, in iteration n, for all $w_i\in S_2$.

In a similar way, we could define the similarity between words:

$$\mathrm{sim}_{n+1}(w_1,w_2) = \Big(\sum_{s\in w_1} \max_{S_j\in w_2} \mathrm{sim}_n(s,s_j) \Big) / k \quad \text{(equation 2)}$$

where k is the number of sentences S which contain word w_1.

2.2. Word Similarity and Sentence similarity.

Suppose that we have a context c containing a polysemous word w that could appear with senses $s_1,s_2,\dots,s_k$. We also have the feedback set (sense-related examples) for each sense of w. Our aim is to disambiguate w.

Two matrices are calculated:

The word similarity matrix WSM_{pxp} where the p rows and p columns keep the words encountered in context c and sense related examples. Each cell (i,j) of the matrix WSM contains a value between 0 and 1, indicating the degree of similarity between words w_i, w_j.

The sentence similarity matrix $SSM_{1\ x\ (r+1)}$ where in the single row is kept the context of w and in the columns all the sense-related examples plus the context of w.

The similarities are calculated by using the Equations (1) and (2) in an iterative process as follows:

First : Initialize word similarity matrix WSM, to the identity matrix, so that each word to be fully similar to itself and fully dissimilar to other words.

Do

Update the sentence similarity matrix *SSM*, using the word similarity matrix *WSM*;
Update the word similarity matrix *WSM*, using the sentence similarity matrix *SSM*;

While *StopCondition*;

In the *StopCondition* the "True" value is assigned when the changes in the similarity values are small enough. [Carov & Edelman 97] give a proof that word and sentence similarity values converge and a detailed description of the stop condition forming a basis for our method.

Let us clarify how the algorithm works in the above example of the three sentences (S_1, S_2, S_3) using a *3x3 SSM*.

In the sentences we have the following words: w1= eat, w2= banana, w3=taste w4=apple. Hence, after the initialization phase the word similarity matrix *WSM* will be:

WSM	w1	w2	w3	w4
w1	1	0	0	0
w2	0	1	0	0
w3	0	0	1	0
w4	0	0	0	1

After the first iteration.

WSM	w1	w2	w3	w4
W1	1	0.75	0.25	0.75
w2	0.75	1	0.75	0.25
w3	0.5	1	1	0
w4	1	0.5	0	1

SSM	S1	S2	S3
S1	1	0.5	0.5
S2	0.5	1	0
S3	0.5	0	1

After the second iteration.

WSM	w1	w2	w3	w4
W1	1	1	0.75	0.94
w2	1	1	0.94	0.75
w3	1	1	1	0.63
w4	1	1	0.63	1

SSM	S1	S2	S3
S1	1	0.88	0.88
S2	1	1	0.63
S3	1	0.63	1

This simple example demonstrates how the similarity values are evolved with the number of iteration and also demonstrates the transitivity of similarity measures, although we used only two iterations. This allows us to capture high-order conceptual relationships. For example, the sentence S2 after the second iteration is similar to S3 with a similarity value 0.63. Note that word similarity and sentence similarity are asymmetric [Carov & Edelman 97].

3. Disambiguation procedure

Suppose that we want to disambiguate a word that appeared in a context (the local textual information around the word, that is, a sentence containing the word or the two adjacent sentences in the case of a small sentence).

In the present work, Internet is used as a resource to gather the essential information for each word in order to disambiguate its appearances. Hence, for each word and for all its senses, we collect sentences that contain nouns found in WordNet definitions (of the sense and its direct hyponyms). These sentences are called the "feedback set" or the "sense-related examples" for our algorithm.

The disambiguation procedure comprises the following steps:

"For_ the_first_ time" procedure.

a) If we do not have any sense related examples, for each sense of the "target" word, we search Wordnet definitions for keywords.

b) Based on the previous search we form the new queries and search the Internet for sense-related examples. After collecting the relevant text, we parse the examples to extract the features of each sentence.

For each target word
 For each sense of the target word
  If sense-related examples are not available
   Perform "*For_ the_first_ time" (procedure)*;
  Else
   c) Form the similarity matrices *WSM* and *SSM* ;
   d) Calculate similarity measure and update *WSM* and *SSM*, in an iterative way, until the stop condition will be satisfied;
   e) Tag the context of the target word finding the maximum similarity based on the final *SSM*;
   g). Add the disambiguated context into the set of sense-related examples;

We describe below the steps of the algorithm:.

a) search WordNet definitions for keywords.

The majority of synsets in WordNet has a definition, known as gloss, that contains defining phrases and (usually) example phrases of the typical use of the sense. More than 95% of synsets have glosses [Rada & Dan 99].

If the synset has a gloss, we tag it using Brill's "part-of-speech tagger" and extract all the nouns. A simple interface with WordNet morphology (function *morphstr()*) returns the basic *lemmas* of the nouns words. These *lemmas* with all the words in the synset are the keywords that will be combined to form a composite searching query.

Example

For the sense *administration#1* the WordNet entry is the following:

Synset: {*administration, disposal*}

Defining gloss: (*a method of tending to (especially business) matters*)

Tagging the gloss with Brill's "part-of-speech tagger" gives the following results:

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

The nouns (and the keywords returned by the *morpstr()* function) are the following: *method*, *business*, *matters*. The possible combinations between synset words and extracted nouns, taken two at a time, give the following composite keywords:

(administration, method), (administration, business), (administration, matter)

(disposal, method), (disposal, business), (disposal, matter).

In the cases where the synsets have not a gloss (5%) we use as keywords only the words in the synset. We also use hyponyms because they have some relevance to the hypernym' sense.

It is typical, when somebody gives a definition (of a noun), to use a "superordinate" (broader) term adding distinguished / specialized features [Miller 93]. This is evident in WordNet.

If we examine the defining glosses of hyponyms for *administration#1* we can find:

=> *line_ management -- (administration of the line functions of an organization; administration of activities contributing directly to the organization's output)*

=> *justice, judicature -- (the administration of law; the act of determining rights and assigning rewards or punishments)*

=> conducting -- (the way of administering a business)

=> *organization, organisation -- (the act of organizing a business or business-related activity)*

=> running -- *(the act of administering or being in charge of something)*

=> polity -- (*shrewd or crafty management of public affairs)*

Working, in a similar way, we extract the nouns from the gloss that contains the "superordinate" term. The composite keywords that we form are all the combinations of the "superordinate" term with each extracted noun and its adjective. For example, from the first hyponym *line_managment* we form the composite keys (*administration, function*), (*administration, organization*).

b) Searching the Internet.

In our method, we use the Altavista full-text search engine to search the web for sense-related examples (texts that contain the keywords from the WordNet definitions specified above).

Altavista stores every word of pages on the Internet in a searchable index and also provides the possibility to create queries using logical operators: *AND*, *OR*, *NOT*, *NEAR*, etc. The *NEAR* operator is very useful for our task because finds documents containing the specified words within 10 words adjancy. Using the *NEAR* operator and composite keywords, we form queries and ask Altavista to search for pages containing these words in a small window of consecutive words. For example, the composite keyword (*administration, line, function, organization*) will give us the query *administration NEAR line NEAR function NEAR organization*.

The collected sense-related examples are tagged using Brill's "part-of-speech tagger", lemmatized by WordNet morphology functions, and the features are extracted. These features are used in the disambiguation procedure as the feedback set.

4. Experimental Results.

We evaluated our method trying to disambiguate the occurrences of 4 polysemous words: *doubt, administration*, *act*, *plant*.
For these words we selected all their occurrences in the tagged texts of Semcor, the semantic concordance files tagged to WordNet 1.6 database, where we knew the senses of the words in the texts and we tried to disambiguate them.
The WordNet senses, for each word in our experiment, and the number of times that each sense is tagged in Semcor are listed in table 1.

word	Senses in WordNet	Tagged Texts	Total
doubt	#1 "the state of being unsure"	26	
	#2 "uncertainty about the truth"	4	30
administration	#1 "disposal"	8	
	#2 "Governing body"	3	
	#3 "Administering medication"	2	
	#4 "The tenure of a president"	1	14
act	#1 "enactment"	35	
	#2 "human activity"	26	
	#3 "a subdivision of a play, opera"	9	
	#4 "a short theatrical performance"	3	
	#5 "a manifestation of insincerity"	0	73
Plant	#1 "industrial plant"	338	
	#2 "living organism"	207	
	#3 "planted secretly by police etc."	2	
	#4 "an actor situated in the audience"	0	547

Table 1. The WordNet senses of the 4 polysemous words.

For each sentence in Semcor we extracted the features, very easily, because the sentence is syntactically tagged (by Eric Brill's "part-of-speech tagger") and the basic lemma for each word is contained in the sentence's SGML-like format. Table 2 shows the extracted features for each sentence in which the word *administration* appears in Semcor files.

administration#1
(jury say believe office achieve efficiency reduce cost)
(jury praise operation atlanta police_department group location location prison_farm group group)
(kill use election reproach)
(coconut contain dessert bring_up problem)
(award executor administrator payment make person person find comptroller_general united_states necessity compliance requirement law with_respect_to estate)
(public_law group federal_government assume responsibility expenditure counseling placement disabled cost provide client rehabilitation case services)
(grant certiorari view importance question)
(initiative control remain school)

administration#2
(maintain facility inventory business_concern register)
(recall liberal person gather utopian san_francisco framework rationalize war rationalize want fear world united_nations)
(strategy blockade center attention official member congress officer pentagon)

administration#3
(emotional_state produce drug influence cortical_potential manner synchrony prevail eeg animal tranquilizer asynchrony application psychoactive_drug)
(continuous recommend cows)

administration#4
(year person red_scare develop country)

Table 2. Extracted features for sentences in Semcor files

For each sense of, the above mentioned, four (4) words we searched the definitions in WordNet and then an Internet search was conducted using Altavista. Eventually, all the sentences containing the words of the composite keywords were collected.
After the “part-of-speech tagging” with Brill’s tagger we extracted the features of these sentences. Applying our algorithm, we calculated the similarity measures and disambiguated the occurrences of the four words in Semcor files assigning in each occurrence the sense of its most similar sentence (in the feedback set).
Table 3 shows a summary of the algorithm’s performance using:
The resulted feedback set without using the definitions of direct hyponyms and the feedback set using synset and direct hyponyms’ definitions. The use of WordNet hyponymy relation leads to richer collections of sense related-examples and facilitates the disambiguation procedure.
Figure 1 depicts how the number of the sense - related examples is increased using the definitions of hyponymy relation and figure 2 illustrates the disambiguation performance.

Word	sense	Sentences to be disambi-guated	Feedback with out hyponymy Relation	% correct per sense	Feedback with hyponymy relation	% correct per sense
administration	#1	8	394	87,50	654	100
	#2	3	2670	100	2990	100
	#3	2	269	100	401	100
	#4	1	397	100	397	100
Total		**14**	**3730**	**96,88%**	**4442**	**100%**
Act	#1	35	810	94,30	882	97,1
	#2	26	203	80,8	1190	92,3
	#3	9	260	88,9	260	88,9
	#4	3	992	100	1054	100
Total		**73**	**2265**	**91%**	**3386**	**94,58%**
Doubt	#1	26	785	92,3	1045	100
	#2	4	940	100	940	100
Total		**30**	**1725**	**96,15**	**1985**	**100**
Plant	#1	338	800	86,62	1434	90,10
	#2	207	782	86,95	1980	92,75
	#3	2	69	100	69	100
Total		**547**	**1651**	**91,19%**	**3483**	**94,28%**

Table 3. Summary of the performance for the 4 test words

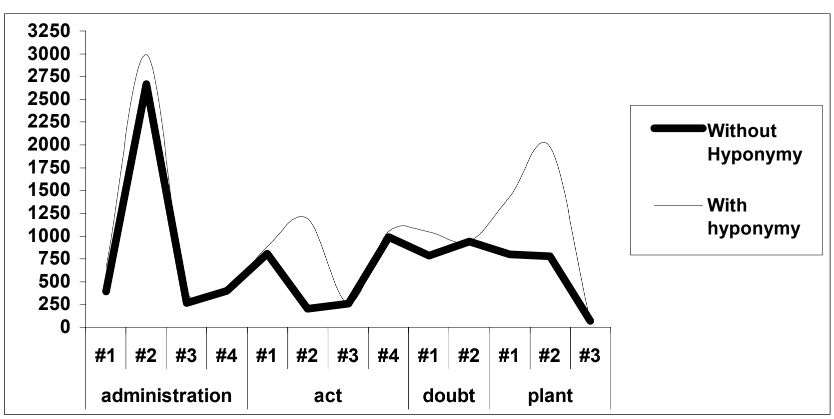

fig 1. Number of sense-related examples collected in the two cases: with the use of definitions of the direct hyponyms and without them.

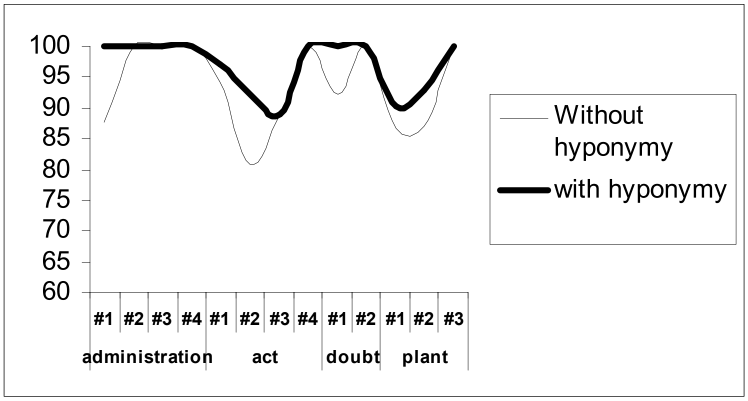

fig 2. % correct disambiguated examples using additionally the definitions of WordNet Hyponymy relation to collect sense-related examples.

5. Discussion and future work.

In this paper, a method for word sense disambiguation is described. This method relies on the sense-related information gathered from Internet for the first time the algorithm is running. When the sense-related information for the senses of a word is collected we could disambiguate, in an easier way, the appearances of the word in a text. At the end of the disambiguation procedure we enrich our collection by adding these disambiguated examples for each sense in the corresponding sense-related information. The use of WordNet is a prerequisite. It provides a definition for each sense giving keywords for searching the Internet to collect the essential sense-related Information. Moreover the WordNet hyponymy relation enables the acquisition of richer collections by using the definitions of direct hyponyms.

WordNet also provides and other semantic relations between synsets, not used here, like meronymy-holonymy and "coordination" for nouns. The usefulness of these semantic relations will be evaluated in a future work.

Our method seems promising for text retrieval. Text Retrieval deals with the problem of finding the relevant documents to a specific query, in a text collection. The parallelism between the relevant documents and the sense-related examples offers us the possibility to apply similar techniques not only to disambiguate word senses but also to match (semantically) related texts in documents and queries.

We also want to evaluate the use of WordNet semantic relations and the proposed similarity based measures in other research areas (e.g. image retrieval technology).

References.

[Aggire & Rigau 96] Agirre E. and Rigau G. (1996). Word Sense Disambiguation Using Cinseptual Density. Proc. 16th International Conference on COLING. Copenhagen.

[Brown et al. 91] Brown, P., Della Pietra, S., Della Pietra, D. and Mercer R. Word sense disambiguation using statistical methods. Proceedings of the 29th Meeting of the Association for Computational.

[Carov & Edelman 97] Karov Y., Edelman S., Similarity-based Word Sense Disambiguation (1997). Association for Computational Linguistics.

[Chlovski & Mihalcea 02] Chklovski T. and Mihalcea R. (2002). Building a Sense Tagged Corpus with Open Mind Word Expert, in Proceedings of the ACL 2002 Workshop on "Word Sense Disambiguation: Recent Successes and Future Directions", Philadelphia, July 2002

[Dagan et al. 94] Dagan, I. et al., 1994. Similarity-Based Estimation of Word Co- occurrence Probabilities. In Proceedings of the 32nd Annual Meeting of the ACL, Las Cruces, pp.272-278.

[Fellbaum 98] Fellbaum, C., WordNet, An Electronic Lexical Database. The MIT Press.
[Gale et al. 92] Gale, W., Church, K. W., and Yarowski, D. (1992). A method for disambiguating word senses in a large Corpus. Computers and Humanities, 26.
[Hearst 91] Hearst, M. A. (1991). Noun homographs disambiguation using local context in large text corpora. Proceedings of the Seventh Annual Conference of the UW Centre for the New OED and Text.
[Leacock & Chodorow 98] Leacock C., Chodorow M. Combining Local Context and WordNet5 Similarity for Word Sense Disambiguation. Wordnet: An Electronic Lexical Database. Cristiane Fellbaum.
[Leacock & et al.93] Leacock, C., Towell, G., and Voorhess, E. M., (1993). Corpus based statistical sense resolution. Proceedings of the ARPA Human Language Technology Workshop, 260-265. San Francisco: Morgan Kauffmann.
[Luk 95] A. Luk. Statistical sense disambiguation with relatively small corpora using dictionary definitions. In Proceedings of the 33rd Meetings of the Association for Computational Linguistics (ACL-95), pages 181-188, Cambridge, M.A., 1995.
[Miller & Teibel 91] Miller G. and Teibel D., A Proposal for Lexical Disambiguation. In Proccedings of DARPA Speech and Natural Language Workshop, 395-399, Pacific Grave, California. February, 1991.
[Miller 93] Miller G. Nouns in WordNet: A Lexical Inheritance System. Five Papers on WordNet Princeton University (Revised August 1993).
[Miller et al. 93] Miller G., Beckwith, R., Fellbaum, C., Gross, D. and Miller, K. Introduction to WordNet: An On-line Lexical Database. Five Papers on WordNet Princeton University (Revised August 1993).
[Mihalcea 02] Mihalcea R. (2002). Bootstrapping Large Sense Tagged Corpora, in Proceedings of the 3rd International Conference on Languages Resources and Evaluations LREC 2002, Las Palmas, Spain, May 2002.
[Mihalcea & Moldovan 99] Mihalcea R. and Moldovan Dan. (1999). Automatic Acquisition of Sense tagged Corpora. American Association for Artificial Intelligence.
[Resnic 95] Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In Proceedings of IJCAI.
[Resnik 92] Resnik, P. (1992). WordNet and distributional analysis: A class-based approach to lexical discovery. Statistically-Based Natural-Language-Processing Techniques: Papers from the 1992 AAAI.
[Sussna 93] Sussna, M. (1993) 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.
[Voorhess 93] Voorhees, E. (1993) Using WordNet to Disambiguate Word Senses for Text Retrieval. SIGIR-93.
[Yarowski 93] D. Yarowsky. One sense per collocation. In Proceedings ARPA Human Language Technology Workshop, pages 266-271, Princeton, NJ, 1993.
[Yarowski 93] Yarowsky D. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In Proceedings of the 14th International Conference on Computational Linguistics (COLING-92), pages 454-460, Nantes, France, 1992.