International Journal on Artificial Intelligence Tools
Vol. 10, No. 1-2 (2001)
© World Scientific Publishing Company

A HIGHLY ACCURATE BOOTSTRAPPING ALGORITHM FOR WORD SENSE DISAMBIGUATION

RADA F. MIHALCEA and DAN I. MOLDOVAN
Department of Computer Science and Engineering
Southern Methodist University
Dallas, Texas, 75275-0122
{rada, moldovan}@seas.smu.edu

Received July 1, 2000
Revised October 27, 2000

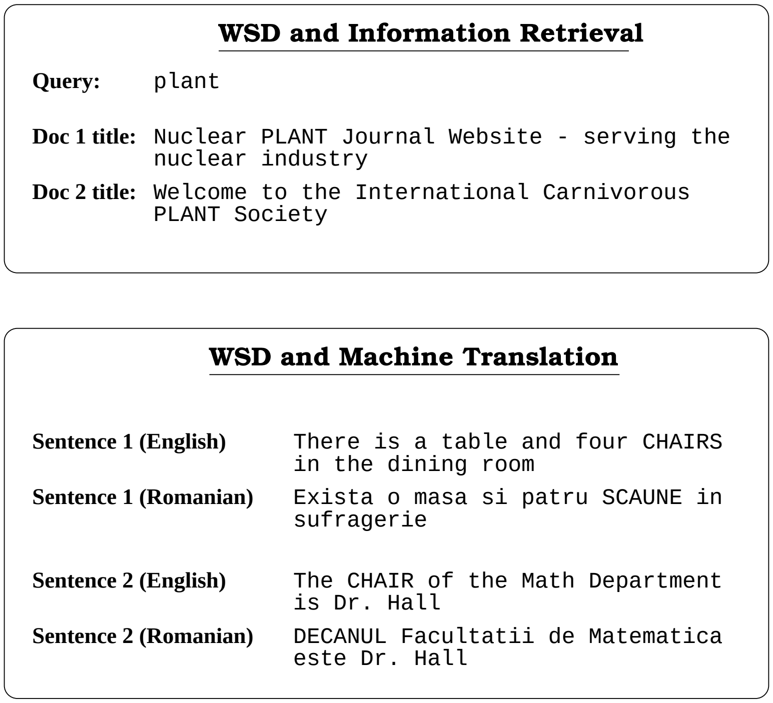
In this paper, we present a bootstrapping algorithm for Word Sense Disambiguation which succeeds in disambiguating a subset of the words in the input text with very high precision. It uses WordNet and a semantic tagged corpus, for the purpose of identifying the correct sense of the words in a given text. The bootstrapping process initializes a set of ambiguous words with all the nouns and verbs in the text. It then applies various disambiguation procedures and builds a set of disambiguated words: new words are sense tagged based on their relation to the already disambiguated words, and then added to the set. This process allows us to identify, in the original text, a set of words which can be disambiguated with high precision; 55% of the verbs and nouns are disambiguated with an accuracy of 92%.

Keywords: Word Sense Disambiguation; Natural Language Processing

1. Introduction

The problem of assigning semantical senses to the words in an open text, known as Word Sense Disambiguation (WSD), is central to many Natural Language Processing applications. Figure 1 illustrates two examples. In Information Retrieval, a search for the word "plant" returns documents containing "plant" meaning "living organism", as well as documents with "plant" meaning "factory". In machine translation, different meanings of the word "chair" have different translations, and thus for a correct translation the correct sense must be known. Other tasks, such as discourse, coreference resolution, summarization and others, can also make use, directly or indirectly of a WSD method. The granularity of word senses is dictated by the dictionary used, as some applications such as machine translation require finer sense disambiguation, and for others, such as information retrieval, a course grain is sufficient.

There are several issues to be considered when evaluating a given WSD al-



WSD and Information Retrieval

Query:	`plant`
Doc 1 title:	`Nuclear PLANT Journal Website - serving the nuclear industry`
Doc 2 title:	`Welcome to the International Carnivorous PLANT Society`

WSD and Machine Translation

Sentence 1 (English)	`There is a table and four CHAIRS in the dining room`
Sentence 1 (Romanian)	`Exista o masa si patru SCAUNE in sufragerie`
Sentence 2 (English)	`The CHAIR of the Math Department is Dr. Hall`
Sentence 2 (Romanian)	`DECANUL Facultatii de Matematica este Dr. Hall`

Fig. 1. Applications of WSD algorithms in other NLP problems.

gorithm: the disambiguation accuracy, the number of words disambiguated, the processing speed and others. Looking back to the WSD history, one notes that the methods developed so far fall in one of the following two categories:

1. WSD methods which can disambiguate only a few preselected words, with very high accuracy (90%+) .

2. WSD methods which can disambiguate all the words (algorithms designed for open text), but with lower precision (70%+, 80%+).

The WSD algorithm presented in this paper achieves a compromise between these two main directions, having the following characteristics:

- it is *designed for open text*: it takes as input a text in English and provides an output in which part of the words are disambiguated;

- it is *semi-complete*, meaning that it is able to assign senses to a subset of the nouns and verbs from the input text; and

- it is *highly precise*, having an accuracy over 90%.

With these characteristics, our algorithm cannot be classified in any of the categories mentioned above. It attempts to disambiguate much more than "a few preselected words", but less than "all the words in the text"; still, the precision achieved is similar with most of the WSD algorithms from the first category.

The algorithm presented here disambiguates the words with respect to WordNet senses. A bootstrapping process starts by creating a set of ambiguous words, initialized with all the nouns and verbs in the text. It then applies several disambiguation procedures and iteratively builds a set of disambiguated words: new words are sense tagged based on their relation to the already disambiguated words. The eight procedures involved in this algorithm are based on machine readable dictionaries (WordNet) and rules acquired from semantic tagged corpora (SemCor).

The results show that 55% of the nouns and verbs are disambiguated with 92% precision. The research reported in this paper is a continuation of our previous work in the field of WSD. [1] [2] [3] [4]

2. Related work

WSD methods can be broadly classified into four types:

1. WSD using information gathered from a lexicon, usually a Machine Readable Dictionaries (MRD), such as WordNet, LDOCE (Longman Dictionary of Contemporary English), Roget's thesaurus, or hand-coded dictionaries. [5] [6] [7] [8] [9] [10] [11]
2. WSD using information gathered from training on a corpus that has already been semantically disambiguated (supervised training methods). [12] [13]
3. WSD using information gathered from raw corpora (unsupervised training methods). [14] [15]
4. WSD methods using machine learning algorithms. [10] [14] [16]

As expected, not all the methods can be classified in one of these categories. There are also hybrid methods that combine several sources of knowledge such as lexicon information, heuristics, collocations and others. [3] [7] [12] [17] .

The method proposed here is a hybrid method, and uses information gathered from a MRD, namely WordNet, and from semantic tagged corpora, i.e. SemCor. It differs from previous approaches in that it uses an iterative approach: the algorithm has as input the set of nouns and verbs extracted from the input text, and incrementally builds a set of disambiguated words. This approach allows us to identify, with high precision, the semantic senses for a subset of the input words. About 55% of the nouns and verbs are disambiguated with a precision of 92%.

The algorithm presented here is part of ongoing research for the purpose of integrating WSD techniques into Information Retrieval (IR) systems[18]. A study of the experiments performed in the NLP community regarding the integration of WSD techniques into IR led us to the conclusion that WSD algorithms can have a positive impact on IR with the condition of being *highly accurate*, even if they are not complete. Having these required characteristics, the algorithm described in this paper proves to be suitable for IR applications. Later in this paper, we investigate the impact of WSD over IR, as reported by different researchers in this field.

This WSD method can also be used in combination with other WSD algorithms with the purpose of fully disambiguating free text. We present the *conceptual density* method as a possibility of increasing the number of words disambiguated to almost all the nouns and verbs in the input text, but with a lower precision.

3. Resources involved in the WSD algorithm

3.1. *WordNet*

WordNet * is a Machine Readable Dictionary developed at Princeton University by a group led by George Miller. [28] WordNet covers the vast majority of nouns, verbs, adjectives and adverbs from the English language. It has a large network of 129,505 words, organized in 99,638 synonym sets, called *synsets.*

The main semantic relation defined in WordNet is the *"is a"* relation; each concept subsumes more specific concepts, called *hyponyms*, and it is subsumed by more general concepts, called *hypernyms.* For example, the concept {*machine*} has the hypernym {*device*}, and one of its hyponyms is {*calculator, calculating machine*}.

WordNet defines one or more senses for each word. Depending on the number of senses it has, a word can be (1) *monosemous*, i.e. it has only one sense, for example the noun *interestingness*, or (2) *polysemous*, i.e. it has two or more senses, for example the noun *interest* which has seven senses defined in WordNet.

3.2. *SemCor*

SemCor is a corpus formed with about 25% of the Brown corpus files; all the words in SemCor are part-of-speech tagged and semantically disambiguated. [29] In the algorithm described here, we use the *brown1* and *brown2* sections of SemCor, containing 185 files; from these, 6 files are used with the purpose of testing our method; the other 179 files form a corpus used to extract rules with Procedure 3 and to determine *noun-contexts* for Procedure 4 (as described in the next section).

3.3. *Evaluation of WSD effectiveness*

A common technique of evaluating a WSD algorithm is to indicate its precision, namely the number of words correctly disambiguated over the total number of words disambiguated. So far, WSD methods attempted to disambiguate either a small number of preselected words or all the words in the text. We are not aware of any algorithms which attempted to disambiguate part of the words in the text,

* WordNet 1.6 is used in our implementation

as our algorithm does; hence, it was not necessary so far to have a measure that specify the percentage of words disambiguated.

We introduce in this paper the notion of *recall*, which is commonly used in IR, but it was not previously used in the WSD field. We define *recall* as the number of words disambiguated over the total number of words attempted to be disambiguated.

Thus, the two measures used to evaluate our WSD algorithm are:

$$precision = \frac{Number\, of\, words\, correctly\, disambiguated}{Number\, of\, words\, disambiguated}$$

$$recall = \frac{Number\, of\, words\, disambiguated}{Number\, of\, words\, attempted\, to\, be\, disambiguated}$$

4. Iterative Word Sense Disambiguation

The algorithm presented in this paper determines, in a given text, a set of nouns and verbs which can be disambiguated with high precision. The semantic tagging is performed using the senses defined in WordNet.

In this section, we present the various procedures used to identify the correct sense of a word. Next, we present the main algorithm in which these procedures are invoked in an iterative manner. We also present the *conceptual density* method, which can be used in addition to the basic procedures such as to increase the recall at the cost of a lower precision.

There is a Set of Ambiguous Words (SAW) which is initialized with all the nouns and verbs in the input text. By applying the procedures described in this section, a Set of Disambiguated Words (SDW) is iteratively build.

4.1. *Procedure 1: Named Entities identification*

Named Entities are words or word sequences which usually cannot be found in common dictionaries, and yet encapsulate important information that can be useful for the semantic interpretation of texts. This procedure uses a Named Entity (NE) component to recognize and identify person names, locations, company names and others. The various names are recognized and tagged. Of interest for us are the PER (person), ORG (group) and LOC (location) tags. The words or word collocations marked with such tags are replaced by their role (person, group, location) and marked as having sense #1.

Example. *"Scott Hudson"* is identified as a person name, thus this word group will be replaced with its role, i.e. *person*, and marked with sense #1.

4.2. *Procedure 2: Monosemous words*

Identify the words having only one sense in WordNet (*monosemous* words). Mark them with sense #1.

Example. The noun *subcommittee* has one sense defined in WordNet. Thus, it is a *monosemous* word and can be marked as having sense #1.

4.3. *Procedure 3: Contextual clues*

With this procedure, we are trying to get contextual clues regarding the usage of a word sense. For a given word W_i, at position i in the text, form two pairs, one with the word before W_i (pair W_{i-1}-W_i) and the other one with the word after W_i (pair W_i-W_{i+1}). Determiners or conjunctions cannot be a part of these pairs. Then, we search for all the occurrences of these pairs found within the semantic tagged corpus formed with the 179 texts from SemCor. If, in all the occurrences, the word W_i has only one sense #k, and the number of occurrences of this sense is larger than a given threshold, then mark the word W_i as having sense #k.

Example. Consider the word *approval* in the text fragment *"committee approval of"*, and the threshold set to 3. The pairs formed are *"committee approval"* and *"approval of"*. No occurrences of the first pair are found in the corpus. Instead, there are four occurrences of the second pair:

"... with the *approval#1 of* the Farm Credit Association ..."
"... subject to the *approval#1 of* the Secretary of State ..."
"... administrative *approval#1 of* the reclassification ... "
"... recommended *approval#1 of* the 1-A classification ..."

In all these occurrences the sense of *"approval"* is sense #1. Thus, *"approval"* is marked with sense #1.

4.4. *Procedure 4: Noun contexts*

The basic idea of this procedure is to determine, for a given noun N in the text, the *noun-context* of each of its senses, and then determine the number of common concepts between each of these *noun-contexts* and the current context of the noun N. This will give us a ranking over the possible senses of N. The *noun-context* is actually a list of nouns which can occur within the context of a given sense #i of the noun N. In order to form the *noun-context* for every sense N_i, we determine all the concepts in the hypernym synsets of N_i. Also, using SemCor, we determine all the nouns which occur within a window of 10 words with respect to N_i.

All of these nouns, determined using WordNet and SemCor, constitute the *noun-context* of N_i. We can now calculate the number of common words between this *noun-context* and the original text in which the noun N is found.

Applying this procedure to all the senses of noun N will provide us with an ordering over its possible senses. We pick up the sense i for the noun N which: (1) is in the top of this ordering and (2) has the distance to the next sense in this ordering larger than a given threshold.

Example. The word *diameter*, as it appears in a text from the aerodynamics field (Cranfield collection), has two senses. The common words found between the *noun-contexts* of its senses and the text are: for *diameter#1*: { property, hole, ratio } and for *diameter#2*: { form}. For this text, the threshold was set to 1, and thus we pick *diameter#1* as the correct sense (there is a difference larger than 1 between the number of nouns in the two sets).

4.5. *Procedure 5: WordNet distance 0 with disambiguated words*

Find words which are semantically connected to the words already disambiguated for which the connection distance is 0. The semantic distance is computed based on the WordNet hierarchy; two words are semantically connected at a distance of 0 if they belong to the same synset. Figure 2 presents a snapshot of the WordNet hierarchy and indicates the semantic distance between the concepts.

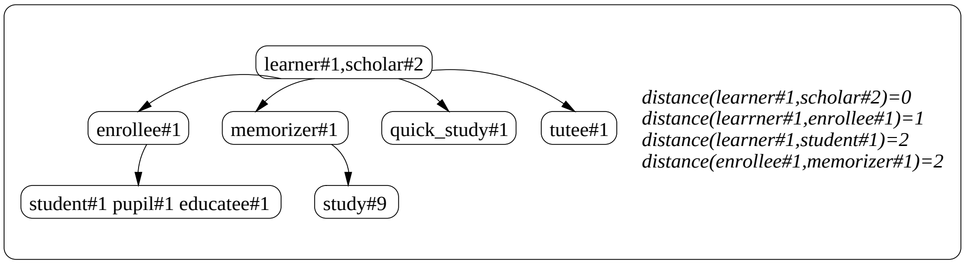

learner#1,scholar#2
enrollee#1
memorizer#1
quick_study#1
tutee#1
student#1 pupil#1 educatee#1
study#9
distance(learner#1,scholar#2)=0
distance(learrner#1,enrollee#1)=1
distance(learner#1,student#1)=2
distance(enrollee#1,memorizer#1)=2

Fig. 2. Distance between concepts measured on the WordNet hierarchy.

Example. Consider these two words appearing in the text to be disambiguated: *authorize* and *clear*. The verb *authorize* is a monosemous word, and thus it is disambiguated with Procedure 2. One of the senses of the verb *clear*, namely sense #4, appears in the same synset with *authorize#1*, and thus *clear* is marked as having sense #4.

4.6. *Procedure 6: WordNet distance 1 with disambiguated words*

Find words which are semantically connected, and for which the connection distance is 0. This procedure is weaker than Procedure 5: none of the words considered by this procedure have been disambiguated yet. We have to consider all the senses of both words in order to determine whether or not the distance between them is 0, and this makes this procedure computationally intensive.

Example. For the words *measure* and *bill*, both of them ambiguous, this procedure tries to find two possible senses for these words, which are at a distance of 0, i.e. they belong to the same synset. The senses found are *measure#4* and *bill#1*, and thus the two words are marked with their corresponding senses.

4.7. *Procedure 7: WordNet distance 0 with ambiguous words*

Find words which are semantically connected to the already disambiguated words, and for which the connection distance is maximum 1. Again, the distance is computed based on the WordNet hierarchy; two words are semantically connected at a maximum distance of 1 if they are *synonyms* or they belong to a *hypernymy/hyponymy* relation.

Example. Consider the nouns *subcommittee* and *committee*. The first one is disambiguated with Procedure 2, and thus it is marked with sense #1. The word *committee* with its sense #1 is semantically linked with the word *subcommittee* by a *hypernymy* relation. Hence, we semantically tag this word with sense #1.

4.8. *Procedure 8: WordNet distance 1 with ambiguous words*

Find words which are semantically connected among them, and for which the connection distance is maximum 1. This procedure is similar with Procedure 6: both words are ambiguous, and thus all their senses have to be considered in the process of finding the distance between them.

Example. The words *gift* and *donation* are both ambiguous. This procedure finds *gift* with sense #1 as being the hypernym of *donation*, also with sense #1. Therefore, both words are disambiguated and marked with their assigned senses.

The procedures presented above are applied iteratively allowing us to identify a set of nouns and verbs which can be disambiguated with high precision. About 55% of the nouns and verbs are disambiguated with 92% accuracy.

4.9. *Conceptual density*

The method presented in this section is meant to be used in addition to the above procedures, in order to obtain a higher recall at the cost of a lower precision. Important applications, such as Information Retrieval, can benefit more from a highly accurate WSD method, even if not complete. But there are other applications which would need a complete disambiguation of the text: using the conceptual density algorithm in addition to the eight procedures presented before allows us to disambiguate almost all the nouns and verbs in the input text (90%), but with lower precision (82%). The basic idea of this method has been first presented in one of our previous papers; it is based on the idea of *semantic density.* [1]

The *semantic density* can be measured by the number of common words that are within a semantic distance of two or more words. The closer the semantic relationship between two words, the higher the semantic density between them. The way it is defined here, the semantic density works well in the case of uniform MRD. In reality, there are gaps in the knowledge representations and the semantic density can provide only an estimation of the actual semantic relatedness between words.

We use the *semantic density* because it is relatively easy to measure it on a MRD like WordNet. This is done by counting the number of concepts two words have in common.

In our previous research, we have used the semantic distance to determine a ranking over the possible combinations of senses of two or more words. In the context of the present research, which addresses WSD from a different perspective, we want to identify the correct sense of a word, rather than a ranking over all possible senses. Thus, we use the semantic distance in a slightly different way: for every noun or verb W which remains ambiguous after applying the procedures above, we determine two words: the word before W (BW) and the word after W (AW), both being already disambiguated (they are part of the SDW set). Then, a metric is used to determine: (1) the semantic distance between BW and W, and

(2) the semantic distance between AW and W. If these two distances provide the same sense for W, then this will be the sense assigned to the word W.

The idea of semantic distance was first introduced by Rada. [31] Since then, several other disambiguation methods based on the distances computed on semantic nets have been developed. [3] [9].

When the semantic distance to be determined is between two words having the same part of speech, this task can be easily accomplished on a MRD like WordNet. The problem arises when the two words involved have different parts of speech, as for example nouns and verbs.

The conceptual density between verbs and nouns seems difficult to determine, without large corpora or without a machine-readable dictionary having semantic links between verbs and nouns. Such semantic links can be traced however if we consider the glosses for the verbs, which are providing a possible context of a verb.

The gloss of a verb synset provides a noun-context for that verb, i.e. the possible nouns occurring in the context of that particular verb. The glosses are used here in the same way a corpus is used.

Given a word W, ambiguous, having the part of speech POS_W, and the word before W, denoted by BW, already disambiguated, having the sense S_{BW} and the part of speech POS_{BW}, the method consists of the following steps:

1. Determine all the possible senses of the ambiguous word W, let us denote them with W_1, W_2 ... W_n

2. For all the senses W_i of the word W, extract the glosses from the synsets in the same hierarchy as W_i.

3. Process the glosses determined in the previous step, i.e. tokenize them and part of speech tag them. [30]

4. For each set of glosses previously determined, extract all the words having the same part of speech as BW, i.e. having the part of speech POS_{BW}. This will result in n sets of words, one for each sense W_i.

5. Determine all the words in the same WordNet hierarchy as the word BW, with its known sense S_{BW}.

6. Determine the common words between the n sets of words determined at step 4, and the set from step 5. Use the metric for semantic distance, as defined below, and determine the conceptual distance C_i between each sense W_i, and BW (with its known sense S_{BW}).

7. The largest value for C_i will indicate #i as the most probable sense for W.

Conceptual Density Metric

The common words between the lists determined at step 4 and 5 above, enable us to determine the conceptual distance between the various senses of the word W and the known sense of BW. The conceptual density is given by the formula:

$$(1) \qquad C_i = \frac{\sum_{k=1}^{|cd_i|} 1}{log(desc_{W_i}) * log(desc_{BW})}$$

where:

- $|cd_i|$ is the number of common concepts between the sets of words from steps 4 and 5 above.

- $desc_{W_i}$ is the total number of words within the set determined at step 4 for W_i

- $desc_{BW}$ is the total number of words within the set determined at step 5 for BW

As words with a larger hierarchy tend to indicate a bigger value for $|cd_i|$, the weighted sum of common concepts has to be normalized with respect to the dimension of the two sets. This is estimated as the logarithm of the total number of words in the sets, i.e. $log(desc_{W_i})$ and $log(desc_{BW})$.

5. Main Algorithm

Step 1. Pre-process the text. This implies tokenization and part-of-speech tagging. The part-of-speech tagging task is performed with high accuracy using an improved version of Brill's tagger. [30] At this step, we also identify the complex nominals, based on WordNet definitions. For example, the word sequence *"pipeline companies"* is found in WordNet and thus it is identified as a single concept. There is also a list of words which we do not attempt to disambiguate. These words are marked with a special flag to indicate that they should not be considered in the disambiguation process. So far, this list consists of three verbs: *be, have, do.*

Step 2. Initialize the Set of Disambiguated Words (SDW) with the empty set SDW={}. Initialize the Set of Ambiguous Words (SAW) with the set formed by all the nouns and verbs in the input text.

Step 3. Apply Procedure 1. The named entities identified here are removed from SAW and added to SDW.

Step 4. Apply Procedure 2. The monosemous words found here are removed from SAW and added to SDW.

Step 5. Apply Procedure 3. This step allows us to disambiguate words based on their occurrence in the semantically tagged corpus. The words whose senses are identified with this procedure are removed from SAW and added to SDW.

Step 6. Apply Procedure 4. This identifies a set of nouns which can be disambiguated based on their *noun-contexts*.

Step 7. Apply Procedure 5. This procedure tries to identify a *synonymy* relation between the words from SAW and SDW. The words disambiguated are removed from SAW and added to SDW.

Step 8. Apply Procedure 6. This step is different from the previous one, as the *synonymy* relation is sought among words in SAW (no SDW words involved). The words disambiguated are removed from SAW and added to SDW.

Step 9. Apply Procedure 7. This step tries to identify words from SAW which are linked at a distance of maximum 1 with the words from SDW. Remove the words disambiguated from SAW and add them to SDW.

Step 10. Apply Procedure 8. This procedure finds words from SAW connected at a distance of maximum 1. As in step 8, no words from SDW are involved. The words disambiguated are removed from SAW and added to SDW.

If it is required to have a complete disambiguation, then the conceptual density method will be used as *Step 11* in this algorithm. The majority of experiments performed so far did not use this method as part of the main algorithm. This is why we present this step as optional.

Step 11(optional). Apply the conceptual density method. This procedure determines the semantic distance between a word W from SAW, and the word before and after W (both already disambiguated, and thus from SDW). In this way, we determine the most probable sense of W, in combination with the word before it, and the most probable sense of W in combination with the word after it. If these two senses coincide, then this will be the sense assigned to W. The words disambiguated in this step are removed from SAW and added to SDW.

6. An Example

We illustrate here the disambiguation algorithm with the help of an example; for this, we consider the following set of sentences extracted from the file br-m02 from SemCor.

"... Instead of inflecting a verb or using an unattached particle to indicate the past or future, Siddo used an entirely different word. Thus, the masculine animate infinitive dabhumaksanigalu'ahai, meaning to live, was, in the perfect tense, ksu'u'peli'afo, and, in the future, mai'teipa. The same use of an entirely different word applied for all the other tenses. Plus the fact that Siddo not only had the normal (to Earthmen) three genders of masculine, feminine, and neuter, but the two extra of inanimate and spiritual. Fortunately, gender was inflected, though the expression of it would be difficult for anybody not born in Siddo. The system of indicating gender varied according to tense. All the other parts of speech: nouns, pronouns, adjectives, adverbs, and conjunctions operated under the same system as the verbs."

First, the text is tokenized and part of speech tagged. We start by initializing

SAW with the set of all nouns and verbs in the text, and SDW is initialized to the empty set. As words are disambiguated using the algorithm described above, they are removed from the SAW set and added to the SDW set.

Using Procedure 1, the complex nominals are identified based on WordNet dictionary and the named entities are recognized. The following complex nominals have been identified: *"perfect tense"* and *"parts of speech"*. Siddo is identified as an organization by the Named Entity recognizer and added to the SDW set.

The monosemous words are identified with Procedure 2, and at this step the SDW set becomes {infinitive#1, perfect_tense#1, tense#1, Earthman#1, neuter#1, part_of_speech#1, pronoun#1}.

Then, we apply Procedure 3, which tries to get rules from SemCor; this will identify *future* as having sense #1; this word is added to SDW. We then apply Procedure 4, which identifies *fact* with sense #1, using its noun-contexts.

Next, we apply Procedure 5, and find another occurrence of the word *future* and assign to this word the correspondent sense, i.e. sense #1. Procedure 6 cannot be applied on this text.

By applying Procedure 7, we try to find words related at a distance of maximum 1 with the words already in SDW. With this procedure, the following words have been disambiguated: *verb#1* (in hypernymy relation with infinitive#1); *past#3* (in hyponymy relation with *tense#1*); *gender#1* (hypernymy relation with *neuter#1*); other two occurrences of *gender* are disambiguated due to the same semantic relation; *noun#2* (hyponymy relation with *part_of_speech#1*); *adjective#2* (hyponymy relation with *part_of_speech#1*); *adverb#1* (hyponymy relation with *part_of_speech#1*); *verb#1* (hyponymy relation with *part_of_speech#1*).

Finally, the SDW set becomes SDW={verb#1, past#3, future#1, infinitive#1, perfect_tense#1, tense#1, Earthman#1, gender#1, neuter#1, part_of_speech#1, noun#2, pronoun#1, adjective#2, adverb#1, verb#1}.

Using this algorithm, we have disambiguated part of the nouns and verbs in the text with high precision. With respect to SemCor, the precision achieved on this text is 92%.

7. Results

To determine the accuracy and the recall of the disambiguation method presented here, we have performed tests on 6 randomly selected files from SemCor. The following files have been used: br-a01, br-a02, br-k01, br-k18, br-m02, br-r05. Each of these files was split into smaller files with a maximum of 15 lines each. This size limit is based on our observation that small contexts reduce the applicability of Procedures 5-8, while large contexts become a source of errors. Thus, we have created a benchmark with 52 texts, on which we have tested the disambiguation method.

In table 1, we present the results obtained for the br-a01 file. The file has been divided into 5 sets of 15 sentences. The number of nouns and verbs considered by the disambiguation algorithm is shown in the first column. For each procedure or pair of similar procedures, we present the number of words disambiguated, as well as the accuracy obtained. Note that the results are cumulative, and thus the last column shows the overall precision obtained for each set. For this file, 55% of the nouns and verbs were disambiguated with 89.3% accuracy.

Table 2 presents the results obtained for the 52 texts created from the 6 SemCor files. The first column indicates the file for which the results are presented; the meaning of the numbers in the other columns is the same as in the previous table.

Table 1. Results obtained for sets of sentences from file br-a01.

	No.	Proc.1+2		Proc.3		Proc.4		Proc.5+6		Proc.7+8	
Set	words	No.	Acc.	No.	Acc.	No.	Acc.	No.	Acc.	No.	Acc.
1	151	35	100%	35	100%	59	94%	73	87.8%	88	82.5%
2	128	47	100%	50	98.4%	65	93.6%	70	90.4%	85	85.7%
3	108	36	100%	40	100%	48	98.2%	53	98.3%	62	91.2%
4	159	40	100%	43	100%	62	89.6%	64	88.6%	72	88.5%
5	159	52	100%	55	100%	76	91.9%	82	91.3%	89	87.9%
6	89	33	100%	35	100%	41	100%	41	100%	43	100%
AVERAGE	132	40	100%	43	99.7%	58.5	94.6%	63.8	92.7%	73.2	89.3%

Table 2. Summary of results for 52 texts.

	No.	Proc.1+2		Proc.3		Proc.4		Proc.5+6		Proc.7+8	
File	words	No.	Acc.	No.	Acc.	No.	Acc.	No.	Acc.	No.	Acc.
br-a01	132	40	100%	43	99.7%	58.5	94.6%	63.8	92.7%	73.2	89.3%
br-a02	135	49	100%	52.5	98.5%	68.6	94%	75.2	92.4%	81.2	91.4%
br-k01	68.1	17.2	100%	23.3	99.7%	38.1	97.4%	40.3	97.4%	41.8	96.4%
br-k18	60.4	18.1	100%	20.7	99.1%	26.6	96.9%	27.8	95.3%	29.8	93.2%
br-m02	63	17.3	100%	20.3	98.1%	26.1	95%	26.8	94.9%	30.1	93.9%
br-r05	72.5	14.3	100%	16.6	98.1%	27	93.2%	30.2	91.5%	34.2	89.1%
AVERAGE	88.5	25.9	100%	29.4	98.8%	40.8	95.2%	44	94%	48.4	92.2%

On average, 55% of the nouns and verbs were disambiguated with 92.2% accuracy. In Table 3, we present the precision and recall of the procedures involved in the main algorithm.

Higher recall can be achieved by using the *conceptual density* method. This will increase the recall to 90%, but the precision will be lower (82%).

Table 3. Precision and recall for the 9 procedures.

Procedure	Recall	Precision
Procedure 1 and 2	29%	100%
Procedure 3	33%	98.8%
Procedure 4	46%	95.2%
Procedure 5 and 6	49%	94%
Procedure 7 and 8	55%	92%

7.1. *Evaluation*

As outlined by Resnik and Yarowsky, it is hard to compare the WSD methods, as long as there are differences in the approaches considered (MRD based methods, supervised or unsupervised statistical methods) and in the words that are disambiguated (words a priori established, only nouns etc.). [32] To our knowledge, there is no other MRD method that disambiguates a subset of the words in the text.

As stated in the introduction to this paper, the methods considered so far are (1) either highly accurate, but work only for a few preselected words, or (2) they are less accurate and are designed to work on open text. In Table 4, we present results which are representative for these two main directions considered in the WSD field. The base line is represented by the simplest WSD method: consider the most frequent sense of a word as the correct one. Translated in WordNet terminology, this implies the assignment of sense #1 to each word in the text. The most accurate WSD methods for open text are those presented by Stetina and Nagao, in 1998, and Mihalcea and Moldovan, in 1999. [3] [13] Finally, the best results to our knowledge in statistical disambiguation are those reported by Yarowsky in 1995. [14]

Table 4. Results obtained by other WSD methods.

	Base line	Stetina 98	Mihalcea 99	Yarowsky 95
noun	80.3%	85.7%	86.5%	93.9%
verb	62.5%	63.9%	67%	-
adjective	81.8%	83.6%	79.8%	-
adverb	84.3%	86.5%	87%	-
AVERAGE	77%	80%	80.1%	-

It is hard to compare our algorithm with any of the algorithms developed so far. This is because the algorithm presented here attempts to disambiguate much more than "a few preselected words", but less than "all the words in the text"; still, the precision achieved is similar with most of the statistical WSD algorithms.

7.2. *Computational time*

Computational time has been always an important issue in the WSD field. Usually, the determination of the right sense of a word is computationally intensive, either because of the preliminary statistics needed for each word, or the explosive combination of senses when considering word pairs, or, as in our case, because of the semantic chains which are built through a large text (usually more than 100 words).

Using our method, on average, about 1 minute is needed to disambiguate 1500 bytes of text (about half page). Considering the fact that the disambiguation of a large collection of texts is a process that can be parallelized, the computational time does not represent anymore a serious limitation.

8. Applications: Word Sense Disambiguation for Information Retrieval

The usage of word senses in the process of document indexing is a pretty much debated issue. The basic idea is to index word meanings, rather than words taken as lexical strings. Experiments performed by different researchers led to various, sometime contradicting results. Nevertheless, the conclusion which can be drawn from all these experiments is that a highly accurate Word Sense Disambiguation algorithm is needed in order to obtain an increase in the performance of IR systems.

Ellen Voorhees tried to resolve word ambiguity in the collection of documents, as well as in the query, and then she compared the results obtained with the performance of a standard run. [19] [20] Even if she used different weighting schemes, the overall results have shown a degradation in the IR effectiveness when word meanings were used for indexing. Still, as she pointed out, the precision of the WSD technique has a dramatic influence on these results. She states that a better WSD can lead to an increase in the IR performance.

A rather "artificial" experiment in the same direction of semantic indexing is provided by Sanderson.[21] He uses pseudo-words to test the utility of disambiguation in IR. A pseudo-word is an artificially created ambiguous word, like for example "banana-door" (pseudo-words have been introduced for the first time by Yarowsky, as means of testing WSD accuracy without the costs associated with the acquisition of sense tagged corpora. [22]). Different levels of ambiguity were introduced in the set of documents prior to indexing. The conclusion drawn was that WSD has little impact on IR performance, to the point that only a WSD algorithm with over 90% precision could help IR systems.

The reasons for the results obtained by Sanderson have been discussed in by Schuetze and Pedersen.[24] They argue that the usage of pseudo-words does not always provide an accurate measure of the effect of WSD over IR performance. It is shown that in the case of pseudo-words, high-frequency word types have the majority of senses of a pseudo-word, i.e. the word ambiguity is not realistically modeled. More than this, Schuetze and Pedersen performed experiments which have shown that semantics can actually help retrieval performance. They reported an increase in precision of up to 7% when sense based indexing is used alone, and up to 14% for a combined word based and sense based indexing.

One of the largest studies regarding the applicability of word semantics to IR is reported by Krovetz. [25] [26] When talking about word ambiguity, he collapses both the morphological and semantic aspects of ambiguity, and refers them as *polysemy* and *homonymy*. He shows that word senses should be used in addition to word based indexing, rather than indexing on word senses alone, basically because of the uncertainty involved in sense disambiguation. He had studied extensively the effect of lexical ambiguity over IR; the experiments described provide a clear indication that word meanings can improve the performance of a retrieval system.

Overall, the conclusion is the we need a highly precise algorithm to be able to increase the quality of the IR systems. Our own experiments performed in the field of WSD for IR have shown that the disambiguation of the words in the collection of documents, prior to indexing, can increase both the precision and recall of an IR system [18].

9. Conclusion

In this paper, we presented a method for disambiguating the nouns and verbs in an input text. The novelty of this method consists in the fact that the disambiguation process is done by bootstrapping. Several procedures, described in the paper, are applied such as to build a set of words which are disambiguated with high accuracy: 55% of the nouns and verbs are disambiguated with a precision of 92.2%.

The most important improvements expected from a WSD method are *precision* and *speed*. In the case of our approach to WSD, we can also talk about the need for an increased *recall*, meaning that we want to obtain a larger number of words which can be disambiguated in the input text.

The precision of more than 92% obtained during our experiments is very high, considering the fact that WordNet is very fine grained and sometime the senses are very close to each other. The accuracy of 92% obtained is close to the precision achieved by humans in sense disambiguation.

As stated earlier in this paper, IR systems can benefit from a WSD method which enables the disambiguation of some of the words with high accuracy. This enables an efficient combined word-based and sense-based combined indexing, without having the errors introduced by a complete disambiguation process with a lower accuracy.

References

[1] Mihalcea, R. and Moldovan, D. Word sense disambiguation based on semantic density. *Proceedings of COLING-ACL '98 Workshop on Usage of WordNet in Natural Language Processing Systems*, Montreal, Canada, 1998.

[2] Mihalcea, R. Word sense disambiguation and its application to the Internet search. Master's thesis, Southern Methodist University, 1999.

[3] Mihalcea, R. and Moldovan, D. A method for Word Sense Disambiguation of unrestricted text *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, pages 152-158, College Park, MD, 1999.

[4] Mihalcea, R. and Moldovan, D. An iterative approach to Word Sense Disambiguation *Proceedings of FLAIRS-2000*, pages 219-223, Orlando, FL, May 2000.

[5] Yarowsky, D. Word Sense Disambiguation using statistical models of Roget's categories trained on large corpora. *Proceedings of COLING-92*, pages 454-469, Nantes, France, 1992.

[6] Miller, G., Chodorow, M., Landes, S., Leacock, C. and Thomas, R. Using a semantic concordance for sense identification. In *Proceedings of the ARPA Human Language Technology Workshop*, pages 240-243, 1994

[7] Bruce, R. and Wiebe, J. Word sense disambiguation using decomposable models. *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL-94)*, pages 139-146, LasCruces, New Mexico, 1994.

[8] Li, X., Szpakowicz, S. and Matwin, M. A Wordnet-based algorithm for word semantic sense disambiguation. *Proceedings of the 14th International Joint Conference on Artificial Intelligence IJCAI-95*, pages 1368-1374, Montreal, Canada, 1995.

[9] Agirre, E. and Rigau, G. A proposal for word sense disambiguation using conceptual distance. *Proceedings of the International Conference "Recent Advances in Natural Language Processing" RANLP'95*, Tzigov Chark, Bulgaria, 1995.

[10] Leacock, C.; Chodorow, M. and Miller, G.A. Using Corpus Statistics and WordNet Relations for Sense Identification, *Computational Linguistics vol.24 no.1*, pages 147-165, 1998.

[11] Siegel, E. Disambiguating Verbs with the WordNet category of the direct object *Proceedings of COLING-ACL '98 Workshop on Usage of WordNet in Natural Language Processing Systems*, Montreal, Canada, July 1998.

[12] Ng, H.T. and Lee, H.B. Integrating multiple knowledge sources to disambiguate word sense: An examplar-based approach. *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL-96)*, pages 40-47, Santa Cruz, 1996.

[13] Stetina, J. and Kurohashi, S., and Nagao, M. General Word Sense Disambiguation method based on a full sentential context. *Proceedings of COLING-ACL '98 Workshop on Usage of WordNet in Natural Language Processing Systems*, Montreal, Canada, July 1998.

[14] Yarowsky, D. Unsupervised word sense disambiguation rivaling supervised methods. *Proceedings of the 33rd Annual Meeting of the Association of Computational Linguistics (ACL-95)*, pages 189-196, Cambridge, MA, 1995.

[15] Resnik, P. Selectional preference and sense disambiguation. *Proceedings of ACL Siglex Workshop on Tagging Text with Lexical Semantics, "Why, What and How?"*, Washington DC, April, 1997.

[16] Towell, G. and Voorhess, E. Disambiguating Highly Ambiguous Words, *Computational Linguistics vol.24 no.1*, pages 1125-145, 1998.

[17] Rigau, G., Atserias, J. and Agirre, E. Combining unsupervised lexical knowledge methods for word sense disambiguation. *Proceedings of joint 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics ACL/EACL'97*, pages 48-55, Madrid, Spain, 1997.

[18] Mihalcea, R. and Moldovan, D. Semantic indexing using WordNet senses. *Proceedings of the ACL Workshop on IR & NLP*, Hong Kong, October, 2000.

[19] Voorhees, E. Using WordNet for text retrieval. *WordNet, An Electronic Lexical Database*. The MIT Press, pages 285-303, 1998.

[20] Voorhees, E. Natural language processing and information retrieval. *Information Extraction: towards scalable, adaptable systems. Lecture notes in Artificial Intelligence, #1714*, pages 32-48, 1999.

[21] Sanderson, M. Word sense disambiguation and information retrieval. *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 142-151, Springer-Verlag, 1994.

[22] Yarowsky, D. One sense per collocation. *Proceedings of the ARPA Human Language Technology Workshop*, 1993.

[23] Voorhees, E. Query expansion using lexical-semantic relations, *Proceedings of the 17th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pages 61-69, Dublin, Ireland, 1994.

[24] Schutze, H. and Pedersen, J. Information Retrieval based on word senses, *in Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 161-175, 1995.

[25] Krovetz, R. and Croft, W.B. Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems*, 10(2):115-141, 1993.

[26] Krovetz, R. Homonymy and polysemy in information retrieval. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97)*, pages 72-79, 1997.

[27] Moldovan, D. and Mihalcea, R. Using WordNet and lexical operators to improve Internet Searches, *IEEE Internet Computing vol.4 no.1*, pages 34-43, 2000.

[28] Fellbaum, C. *WordNet, An Electronic Lexical Database.* The MIT Press, 1998.

[29] Miller, G.A., Leacock, C., Randee, T. and Bunker, R. A Semantic Concordance. *Proceedings of the 3rd DARPA Workshop on Human Language Technology*, pages 303-308, Plainsboro, New Jersey. 1993.

[30] Brill, E. A simple rule-based part of speech tagger. *Proceedings of the 3rd Conference on Applied Natural Language Processing*, Trento, Italy, 1992

[31] Rada, R., Mili, H., Bickell, E. and Blettner, B. Development and Application of a Metric on Semantic Nets. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, pp 17-30, Jan/Feb 1989.

[32] Resnik, P. and Yarowsky, D. A perspective on Word Sense Disambiguation methods and their evaluation. *Proceedings of ACL Siglex Workshop on Tagging Text with Lexical Semantics, Why, What and How?*, Washington DC, April, 1997.