Using Conditional Probabilities of Weighted Terms for a Lexicon Based Sense Disambiguation System.

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: - In this paper we propose and discuss a method for Word Sense Disambiguation. A Lexicon approach is presented based on the use of the WordNet. More precisely, the context and the senses of the ambiguous word are represented as vectors of weighted terms, in a vector space model, using WordNet definitions and the rich hypernymy relations. Calculating the conditional probabilities (relative frequencies) for these terms we can measure the similarity of the target word with a sense. Hence, the ambiguous word in the context is assigned to the most similar sense. Our algorithm does not need any training and is tested on the entire Semantic Concordance Corpus (Semcor). The estimated performance of the algorithm is 78,13% .

Key-Words: - Word Sense Disambiguation ; Natural Language Processing ; Wordnet ; Vector space model

1 Introduction

The task of resolving the word ambiguity in a context is generally known as Word Sense Disambiguation (WSD). It is widely used in all Natural Language Processing systems. WSD is also applied to Machine translation, Information Retrieval, parsing etc. The WSD problem is, in general, a very difficult one and all the efforts rely on the examination of the context in which the target word[1] occurs. We can distinguish two different approaches: Supervised Learning methods and unsupervised ones. In a supervised approach the system learns to disambiguate during a training phase based on previously (manually) disambiguated text corpora [10], [8], [4], [5], [21], [13]. Alternative methods rely on the definitions of senses in dictionaries and / or thesauri [22], [23], [20], [21], [1].

In unsupervised approaches, lexical (training) resources are not necessary, especially, when dealing with information from specialized domains. The proposed algorithms usually cluster some contexts of an ambiguous word into a number of groups and then try to disambiguate. [21], [4], [5].

The use of Lexicons, in the supervised approaches, is usually based on the Lesk's method [15]. Lesk believes that dictionary definitions of the ambiguous word are good "indicators" for the senses (of the word). According to this method, each lexicon definition is represented as a *bag* of words occurring in the definition. A bag is a set of words in which all the structure and the linear or hierarchical ordering of words within the context is ignored. For an ambiguous 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 target word) their definitions are retrieved and another bag, the context bag, containing all the words occurring in the definitions is also 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. Another use of lexicon is that of the Longman's Dictionary of Contemporary English (LDOCE). Cowie [7] proposed a similar method using definitions from this lexicon and improved the results applying a procedure of simulated annealing.

1.1 The use of WordNet.

WordNet is an electronic lexical database developed at the University of Princeton [16]. Lexical entries are organized around the concept of a synset. Each synset consists of one or more words that are considered to be identical in meaning, together with a definition (the gloss), which defines that meaning. The gloss comprises the defining part and usually one or more defining examples. Synsets can be inter-related by one or more predefined relations. WordNet supports two types of relations: semantic relations, such as hypernymy, hyponymy, meronymy, holonymy, troponymy etc. which link concepts (i.e. synsets), and lexical relations, such as antonymy, which links individual words. In this work we use the synset definitions and the Hypernymy/hyponymy relation. Let us present these two WordNet aspects by an example.

If we look up the word administration in the dictionary the system responds:

[1] By the term target word we mean the word that is to be disambiguated.

The noun administration has 4 senses (first 4 from tagged texts)

1. administration, disposal -- (a method of tending to (especially business) matters)

2. administration, governance, establishment, brass, organization, organisation -- (the persons (or committees 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")

3. administration, giving_medication -- (the act of administering medication)

4. presidency, presidential_term, administration -- (the tenure of a president; "things were quiet during the Eisenhower administration")

The word administration has 4 senses (synsets) each of which consists of the synset words and the gloss (the defining part and usually the defining examples). These synsets interrelate with other synsets in WordNet by a number of semantic relations. Table 1 shows the hyponyms and the hypernyms of the word administration sense #2 (see the synset administration, governance, establishment, brass, organization, organization).

Today, research works in WSD, usually, pertain WordNet. We mention some remarkable results. Lee et al. [14] and Leacock and Chodorow [13] propose a measure of *semantic similarity* by calculating the length of the path between the two nodes in the hierarchy. Agirre and Rigau [1] propose a method based on the *conceptual distance* among the concepts in the hierarchy and provide a conceptual density formula for this purpose. Budanitsky and Graeme [6] present some experimental results comparing the above systems in a real-word spelling correction system. Voorhess [23] is dealing with the problem of the lack of "containment" of clear divisions in the WordNet hierarchy and it defines some categories. Resnik [22] disambiguates noun instances calculating the (semantic) similarity between two words and choosing the most informative "subsumer" (ancestor of both the words) from an IS-A hierarchy.

In this work we use WordNet to calculate semantic similarity in a different way inspired by document and text retrieval techniques [19], [11], [12]. We represent the context and each sense of the target word as vectors in a multi-dimensional space. Parsing the WordNet definitions and extracting the contained words (terms) we construct vector entries. Each vector contains a weight for every term that is an estimation of the importance of that term in the disambiguation procedure. This estimation is based on the relative depth of the synset, whose definition contains the term, within the WordNet taxonomy. Such an approach based on the ordinary *vector space model* could enable us to use the large class of the existing semantic similarity measures. But our algorithm is dealing with weighted terms. Hence, we chose to use probabilistic similarity measures. Another reason for this choice is that an initial experimentation with the common *cosine (*measure) [19] had a low performance. To work with probabilistic similarity measures we convert the entries of a vector into conditional probabilities (or relative frequencies) by dividing (each of) them by the sum of all the entries.

The use of weights for the terms in vector space models is a well-known technique [19]. A weighting scheme using WordNet' s relations is proposed by Sussna in [20]. The disambiguation procedure is based on the use of a semantic distance between topics in WordNet. The synonymy relation gets a weight of zero value and the hypernymy, hyponymy, holonymy and meronymy relations are assigned weights in the range [1, 2]. Antonymy arcs are assigned the value 2.5. Ganesal et al. [9] propose new similarity measures that exploit a hierarchical domain structure in order to produce more intuitive similarity scores. The proposed similarity measures could be applied to calculate semantic similarity between vectors exploiting the WordNet hierarchy. Instead of this, we adopt a simpler way (to exploit WordNet hierarchy), but in the same direction of capturing the semantic distance between the different levels of the hierarchy. Eventually, in this paper, a word term is assigned a weight that is inversely proportional to the hierarchy depth of the synset' s definition. In the next section 2, an overview of the WordNet and probabilistic similarity measures is given offering a basis for problem formulation. Then a detailed presentation of our method follows in sections 3 (definition of the semantic similarity and description of the method). Evaluation results are given in section 4. Related word, further discussion of our method and references are also given.

2. Problem Formulation

A naive approach to the problem formulation can be based on a simple example. Suppose that we want to disambiguate an occurrence of the word "*administration"* in the following context.

However, the jury said it believes these two offices should be combined to achieve greater efficiency and reduce the cost of administration.

To represent the context and the senses in the *vector space model* we can form five vectors: One vector c, for the context, containing all "surrounding" words of the target word *administration* and four vectors s_1,s_2,s_3,s_4 for the four senses of the target word. These vectors are shown below.

C=(*However the jury said it believe this two office should be combine to achieve great efficiency and reduce the cost)*

S_1=*(a method of tend to especially business matter).*

S_2=*(the person or committees or department who make up a govern body and who administer something).*

S_3=*((the act of administer medication).*

S_4=*(the tenure of a president).*

No *stop list* of words is used to remove the most frequent words. Only words with length less than 3 are omitted. Some morphological analysis is also applied to handle all the inflectional forms of a word. After a *part_of_ speech tagging* the above vectors are expanded using in the same way: a) the definitions of all nouns and verbs occurring in the vectors and b) the definitions of all associated (with these nouns and verbs) *synset hypernyms* in the WordNet taxonomy.

At the final stage of calculating the similarity. Instead of words we prefer to use (their) weights. These could be estimated using the hypernymy relation. After the vectors' expansion with terms from hypernymy relation and the assignment of a weight in each term, we can construct an *inverted index* that lists for each word all vectors that contain an entry for it (its weight). Hence, we can proceed to calculate maximum likelihood estimations for each word.

2.1 Semantic Similarity.

In a vector space model, the *inverted index* is a matrix A and each entry a_{ij} contains the weight (usually the number of occurrences) for a term j that occurs in the vector i. If we suppose that the first row stands for the context C and the remaining rows stand for senses S_i then each entry a_{ij} will contain the weight of the term j that occurs in the context vector C when $i=0$, or in the sense vector S_i when $i\neq 0$. Our aim is to calculate similarity between the context C and each sense vector S_i, counting not simple occurrences for the various words, but using WordNet to estimate the most appropriate weight for each entry of the matrix.

Before the presentation of the weighting scheme we, shortly, describe the semantic similarity measure used. Let us suppose that we have a matrix A and each entry contains counts of occurrences for a term. Instead of this count an appropriate weight for the term can be inserted into the corresponding matrix entry. Hence, we convert this matrix into a matrix of conditional probabilities by dividing each new entry in a row by the sum of all entries in the same row. For example, a portion of matrix A is shown in table 2

The entry (*C, committee*) is 2 and is transformed into

p(committee | C)=2/4=0.5.

Calculating all the conditional probabilities we take the following distributions:

C:	0.25	0.5	0	0.25
S_1:	0	0.5	0.5	0
S_2:	0.333	0	0.667	0
S_3:	0.5	0	0.5	0
S_4:	0	0.333	0	0.667

An appropriate (dis) similarity measure can be applied to these probability distributions. The most common in use (dissimilarity measure) is the Kullback-Leibler divergence or KL divergence.

This metric (measure) is given by the following equation:

$$D(p \,||\, q) = \Sigma_{x\in X} p(x) \log(p(x)/q(x)). \quad (1).$$

Where p(x), q(x) are probability distributions.

We can think of KL divergence as the 'distance' between two probability distributions.

This metric suffers from two problems in our case:

It gets a value of ∞ when q is zero and is also asymmetric, $D(p||q) \neq D(q||p)$. *Information radius* (IRad) (or total information to the average) overcomes these drawbacks: It is symmetric and presents no problem with infinite values. IRad is given by the equation:

$$IRad(p,q) = D(p||(p+q)/2) + D(q||(p+q)/2) \quad (2)$$

where the notation *D(.|| .) denotes* the KL Divergence defined in eq. (1).

The terms that do not appear in the context vector (as not contributing to the similarity measure) are eliminated. Under this assumption, *IRad* divergence can be used to estimate similarities.

In the next section, we describe the disambiguation method and the way of assigning weights to terms using the WordNet hypernymy relation

3. The disambiguation method.

As we have mentioned, a WordNet sense definition (gloss) consists of the synonyms, the defining part and usually some defining examples. Here in our algorithm we use only defining parts.

Before the term extraction phase a defining part and the context enter into a preprocessing phase that includes part_of_speech tagging, tokenization and stemming. To disambiguate an instance of the target word we form the vectors in the following way:

Based only on the defining part we extract the terms and form a multi-dimensional vector for each sense definition of the target word.

For each word of the context we look up all its senses in the WordNet and, merging all the definitions, we form the context bag.

Then, these vectors are expanded with terms that are extracted by the definitions of all the nouns and verbs contained within the vectors as well as by their hypernym definitions.

3.1 Preprocessing the data.

All the definitions and the context have to pass the preprocessing phase and then they can 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 [3]. 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]
```

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++.

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 by the WordNet team.

Having the forms, we look up WordNet for their hypernyms and taking the glosses, we extract the terms to form the vectors.

Before inserting a term into a vector, a word stemming procedure is applied for removing the inflectional endings.

We decided to use stemming for three reasons: first, with stemming we obtain better disambiguation results, second, to keep the vector sizes and the processing time small and third, to keep compatibility with other text processing applications. An implementation of the widely known Porter Stemming Algorithm [18] used. shape.1 in appendix shows the basic components that represent the stages we follow to create a vector from definitions. The hypernym definitions are extracted from WordNet using the Hypernymy relation for each noun and verb. Then, they are used to expand the vectors. These additional terms are given weights. Terms extracted from lower levels of the WordNet taxonomy take precedence over terms extracted from higher levels. We describe now the way of assigning weights to terms.

3.2 Exploiting the hierarchy and assigning weights into terms.

Suppose we want to disambiguate an instance of a target word in a context. This word is appearing in the same context with other words. It is also appearing in various WordNet synsets (its senses). At the beginning, a weight 1 is assigned to all the synsets in which each context word appears and to all the synsets in which the target word belongs to. Such synsets are called base synsets. Then, we parse the definitions of the base synsets and assign the same weight 1 to the extracted words. If some of these words are nouns or verbs, we look up the WordNet for their hypernyms and going up the hierarchy, at each level, we assign to the hypernymy synsets a weight inversely proportional to its distance from the base synset.

Example

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 ambiguous word. 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 given a weight 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):

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

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

Now, we completed the first level handling the noun "method", occurring in the base synset's definition, and then 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.

Hence, using the way described above, each word (term) is assigned a weight. Terms that are extracted from definitions at lower levels of the hierarchy are assigned a higher weight as compared to terms extracted at higher levels of the hierarchy.

Then, this weight is inserted into the vector-word matrix in the corresponding (i, j) entry (i vector, j term). We must emphasize here that a term weight is inserted into a vector only once. If a term appears repeatedly during a vector expansion only the weight of the first occurrence is taking into account. Let see an example from table 2. This table shows that the term "committee" occurs 2 times in the context C and hence has a count (frequency) of 2. Let us suppose that, the first occurrence has a weight of 1 (that is, it belongs to a base synset definition) and the second occurrence has a weight of 0.5 (that is, it belongs to the second level of hierarchy). Then, the weight 1 is assigned to the term "committee" instead of using two weights.

Hence, this weighting scheme is simple and closely intuitive to the notion of "distance" in a hierarchical domain [see Ganesal et al. 03].

3.3 Assigning the correct sense.

Applying the weighting scheme to all extracted terms, the entries of the inverted matrix have now new balanced values. We divide the weight of its term by the term frequency and then we raise it to 2. We calculate the conditional probabilities and apply the cosine similarity for each pair (C,Si), i=1..N, where N is the number of senses. A sense is assigned to the target word as the correct one, if it is the sense whose vector Si is most similar to vector C. If the same similarity value corresponds to two vectors Si the target word is characterized as ambiguous.

3.4 Related work and evaluation.

We evaluated our method using all the files of the Brown Corpus (Semcor files) and the results were promising. There are algorithms [1], [17], for disambiguating nouns' occurrences, that have also used hypernymy relations (the WordNet, in general) and Semcor – files. They reported a performance about 50%.
Table 3 shows the overall accuracy of our method when evaluating using the Brown1 and Brown2 corpus.

Semcor Files.

% performance	Correct	Incorrect	Ambiguous
Brown1	79.47	20.53	0
Brown2	76.43	23.57	0
TOTAL	**78.13**	21,87	0

Table 3.0

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 preferd 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 [17], 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%.

4. Conclusion

In this work the WordNet is used to disambiguate a polysemous word that appears in a context. Both senses Si and context C are represented as vectors of weighted terms. The weights are calculated using the WordNet hypernymy relation. We do not use the cosine similarity measure to calculate similarities between vector. Instead of this, the vectors are converted into vectors containing conditional probabilities, or relative frequencies, in a vector-term matrix representation. IRad (information Radius) is also used.
As an alternative, we could use normalized vectors. We call a vector normalized if it has unit length according to the Euclidean norm:
Let x=(x1,x2,…, xn) be a vector. Then for normalized vectors:

$$|x| = \sum_{i=0}^{n} x_i^2 = 1$$

To normalize vectors we work as follows:
We multiply the dimensions of each vector by the factor $(1/S)^{1/2}$, where S is the sum of all squared dimensions of the vector.

The cosine similarity measure can be calculated as follows:

$$Cos(x,y) = \frac{x\ y}{|x|\ |y|} = \frac{\sum_{i=1}^{n} x_i y_i}{(\sum_{i=1}^{n} x_i^2)^{1/2} (\sum_{i=1}^{n} y_i^2)^{1/2}}$$

and for normalized vectors it is:

$$Cos(x,y) = \frac{x\ y}{|x|\ |y|} = \sum_{i=1}^{n} x_i y_i$$

We will evaluate our algorithm using this metric in the future.

The way of assigning weights with the hypernymy relation is an aspect that may be improved [9]. The use of hyponymy relation can follow such a direction.

In this work we disambiguated only nouns from their contexts and the results were very encouraging for further improvement, given that we used only a WordNet semantic relation. Eventually, the inclusion of other parts of speech (terms), like adjectives and adverbs in the vector expansion procedure, as well as the disambiguation of (ambiguous) verbs and adjectives and the evaluation of the method on SENSEVAL data will be the matter of future work.

References:

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

[2] Banerjee S. and Pedersen T. (2002). An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In the Proceedings of the Third International Conference on

[3] Brill E. (1992). A simple rule based part of speech tagger. In Proceedings of the Third Conference on Applied Natural Language Processing, ACL, 1992.

[4] Brown, P., Della Pietra, S., Della Pietra, D. and Mercer R. (1991). Word sense disambiguation using statistical methods. In Proceedings of the 29th Meeting of the Association for Computational Linguistics (ACL-91), pages 264-270, Berkley, C.A., 1991.

[5]Brown, P., Della Pietra, S., Della Pietra, D. and Mercer R. (1991). Word sense disambiguation using statistical methods. In Proceedings of the 29th Meeting of the Association for Computational Linguistics (ACL-91), pages 264-270, Berkley, C.A., 1991.

[6] Budanitsky A., Graeme H. (2001). Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. Workshop on WordNet and Other Lexical Resources, in the North American Chapter of the Association for Computation Linguistics (NAACL-2000), Pittsburgh, PA, June 2001.

[7] Cowie J., Cuthrie J. and Guthrie L. (1992). Lexical disambiguation using simulated annealing. Proc. DARPA Workshop on Speech and Natural Language. 238-242. New York.

[8] Gale W., Church W. and Yarowski D. (1992). A method for disambiguating word senses in a large Corpus. Computers and Humanities, 26.

[9] Ganesan F., Garsia-Molina H., Widom J. 03] Exploiting Hierarchical Domain Structure to Compute Similarity. In ACM Transactions on Information Systems, Vol. 21, No. 1, Jannuary 2003, Pages 64-93.

[10] Hearst A. (1991). Noun homographs disambiguation using local context in large text corpora. In Proceedings of the Seventh Annual Conference of the UW Centre for the New OED and Text Research: Using Corpora, 1-22. Waterloo,Ontario, Canada: UW Centre for the New OED and Text Research.

[11] Karanikolas N. and Skourlas C. (2000). Computed Assisted Information Resources Navigation. Medical Informatics and the Internet in Medicine, volume 25, No 2.

[12] Karanikolas N., Skourlas C., Christopoulou A. and Alevizos T. (2003) Medical Text Classification based on Text Retrieval techniques. MEDINF 2003. October 9 - 11, Craiova, Romania.

[13] Leacock C., Chodorow M. (1998). Combining Local Context and WordNet5 Similarity for Word Sense Disambiguation. In Wordnet: An Electronic Lexical Database, pages 265-283. MIT Press, Cambridge MA, 1998.

[14] Lee J. H., Kim H. and Lee Y. J. (1993). Information retrieval based on conceptual distance in IS-A hierarchies. Journal of Documentation, 49(2): 188-207.

[15] Lesk M. (1986). Automatic sense disambiguation: How to tell a pine cone from an ice cream cone. In Proceedings of the 1986 SIGDOC Conference, Pages 24-26, New York. Association of Computing Machinery.

[16] Miller G., Beckwith R., Fellbaum C., Gross D. and Miller K. (1993). Introduction to WordNet: An On-line Lexical Database. Five Papers on WordNet Princeton University.

[17] Montoyo A. and Palomar M. (2001). Specification Marks for Word Sense Disambiguation: New Development. A. Gelbukh (Ed.): CICLing 2001, LNCS 2004. Pp. 182-191, 2001.

[18] Porter M. F. (1980). An algorithm for suffix stripping Sparck Jones, Karen, and Peter Willet, 1997, Readings in Information Retrieval, San Francisco,Morgan Kaufmann, ISBN 1-55860-454-4.

[19] Salton G. and Buckley C. (1988). Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. 24, 5, 513-523.

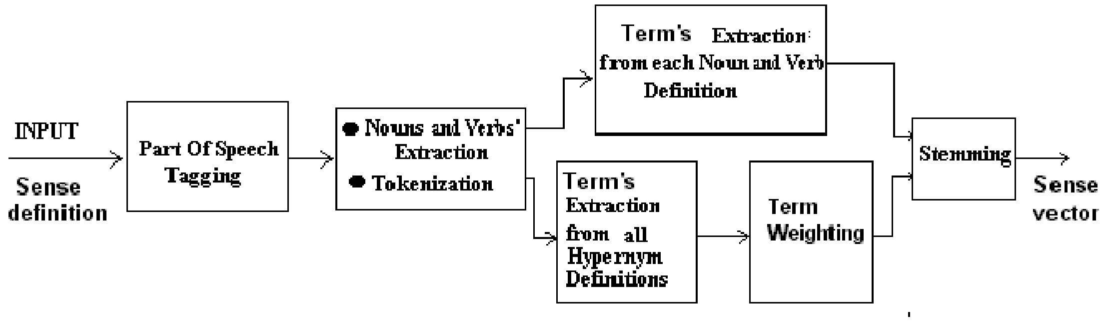
[20] Sussna M. (1993) Word sense disambiguation for free-test indexing using a massive semantic network. In Proceedings of the 2nd International Conference on Information and Knowledge Management. Arlington, Virginia, USA.

[21] Yarowsky D. (1993). 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.

[22] [Resnic 95a] Resnik P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In Proceedings of IJCAI.

[23] [Voorhess 93] Voorhees E. (1993) Using WordNet to Disambiguate Word Senses for Text Retrieval. In Proceedings of 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 171-180, Pittsburgh, PA.

Appendix.



Shape 1. Basic components of the Term Extraction System

<table>
<tr><td colspan="2">administration#2 (#2 means the second sense of the word in Wordnet)</td></tr>
<tr><td colspan="2">“governing body who administers something”</td></tr>
<tr><td colspan="2">Synset :
(administration, governance, establishment, brass, organization, organisation)</td></tr>
<tr><td colspan="2">Gloss
2. administration, governance, establishment, brass, organization, organisation -- (the persons (or committees 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").</td></tr>
<tr><td>Hyponyms</td><td>Hypernyms</td></tr>
<tr><td>=> 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)</td><td>⇨ 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).</td></tr>
<tr><td colspan="2">TABLE 1: The synsets, the gloss, the hyponyms and the hypernyms of the sense administration#2</td></tr>
</table>

	Method	*committee*	*medication*	*president*
C	1	2	0	1
S_1	0	1	1	0
S_2	1	0	2	0
S_3	1	0	1	0
S_4	0	1	0	2

Table 2. A portion of the matrix *A* from the example of *Administration*.