STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **XLIX**, Number 2, 2004

WORD SENSE DISAMBIGUATION BY MACHINE LEARNING APPROACH: A SHORT SURVEY

DOINA TĂTAR

Abstract. There is a renewed interest in word sense disambiguation (WSD) as it contributes to various applications in natural language processing. Applications for which WSD is potentially an issue are: Machine Translation, Information Retrieval (IR), QA systems, Dialogue systems,etc. In this paper we survey vector-based methods for WSD in machine learning approache.

1. Introduction

In the last ten years there has been a dramatic shift in computational linguistics to statistical learning methods (or corpus -based methods). This popularity of statistical methods has its origin in the growing availability of big machine-readable corpora and dictionaries. Some concrete publication statistics illustrate the extent of the revolution in NLP: as an example 63.5 % of the papers in ACL'97 proceedings and 47.7% of the papers in the journal Computational Linguistics in 1997 concerned corpus -based methods, compared with 12.8% and 15.4% in 1990. The argument for a statistical learning approach is to be able to interact successfully with uncertain and incomplete linguistic information. On the other hand natural language can provide machine learning with a variety of interesting and challenging problems such as very large feature space or very large training sets.

In this paper we follow a "machine learning" approaches categorization of WSD as: supervised, bootstrapping and unsupervised (sections 2,3,4). A "machine readable dictionary" based approach of WSD is presented in section 5. Some conclusions about Senseval 3 contest, developed in Marts -April 2004, where we participated with a team for Romanian language [15], will be formulated (section6).

Received by the editors: September 2004.

2000 *Mathematics Subject Classification.* 68T50,68Q32.

1998 *CR Categories and Descriptors.* I.2.7 [**Computing Methodologies**]: Artificial Intelligence – *Natural Language Processing*; G.3 [**Mathematics of Computing**]: Statistical Computing .

2. Machine learning approach in WSD

2.1. **The polysemy.** Word sense disambiguation is the task of assigning sense labels to occurrences of an ambiguous word. This problem can be divided into two subproblems [14]: sense discrimination and sense labeling. Word sense discrimination is easier than full disambiguation since we need only determine which occurrences have the same meaning and not what the meaning actually is.

In many applications full disambiguation is needed as for example in the machine translation. In the following we mean by WSD usually both discrimination and labeling of ambiguous words.

WSD has been a research area in NLP for almost the begin of this field due to the phenomenon of *polysemy* that means multiple related meanings with a single word. At least 40 % of semanticaly signifiant words are ambiguous. Also the problem of WSD is AI complete (that means its solution requires a solution to all the general AI problems of representing and reasoning about arbitrary) and it is one of the most important open problems in NLP [6].

2.2. **Meaning and context.** The systems in the supervised learning approach category are trained to learn a classifier that can be used to assign a yet unseen example to one of a fixed number of senses. That means we have a *trained* corpus, where the system learns the classifier and a *test* corpus which the system must annotate. So, supervised learning can be considered as a classification task, while unsupervised learning can be viewed as a clustering task. Word sense disambiguation (for polysemic words) is the process of identifying the correct sense of words in particular contexts. The precise definition of a sense is a matter of considerable debate within the community. However one would expect the words closest to the target word to be of greater semantical importance than the other words in the text. On the other hand, if two words frequently occur in similar context we may assume that they have similar meanings. The context is hence a source of information and is the only means to identify the meaning of a polysemous word.

Context is used in two ways: a) as *bag of words*, without consideration for relationships to the target word in terms of distance, grammatical relations,etc; b) with relational information. The *bag of words* approach works better for nouns than verbs but is less effective than methods that take other relations in consideration. Studies about syntactic relations determined some interesting conclusions: verbs derive more disambiguation information from their objects than from their subjects, adjective derive almost all disambiguation information from the nouns they modify and nouns are best disambiguated by directly adjacent adjectives or nouns [6].

2.3. **Vector Space Model.** In the following we will use the Vector Space Model (VSM)[9]: a context c is represented as a vector $\vec{c}$ of some features. The definition and the numbers of these features depend on the method selected. A common denominator between the methods is that they excavate information using co-occurrence and collocation statistics. The famous dictum "meaning is use" means that to understand the meaning of a word one has to consider its use in the frame of a concrete context. Context size can vary from one word at each side of the focus word to a more "window" or even the complete sentence. The notations used are:

- $s_1, \cdots, s_{Ns}$ the senses for w;
- $c_1, \cdots, c_{Nc}$ the contexts for w;
- $v_1, \cdots, v_{Nf}$ the features selected (or terms).

In generally, a number of most frequently used words are selected for use as features $v_1, \cdots, v_{Nf}$. When these features have a specific position located to the left and/or the right of the target word w they are *collocational* features, when we ignore the exact position of a feature, we call it a *cooccurrence* feature.

As example we can associate to a context c the vector $\vec{c}$:

- $\vec{c} = (w_1, \cdots, w_{Nf})$ where w_i is the number of times the word v_i occurs in context c;
- $\vec{c} = (w_1, \cdots, w_{Nf})$ where w_i is 1 if the word v_i occurs in context c, or 0 otherwise;
- $\vec{c} = (\cdots w_{i-1}, w_{i+1} \cdots,)$ where w_{i-1} (w_{i+1}) is 1 if the word v_i occurs in context c at the left (right) of the word w or 0 otherwise ;
- $\vec{c} = (\cdots w_{i-k}, w_{i-(k-1)}, ..., w_{i-1}, w_{i+1}, ..., w_{i+k} \cdots,)$ where w_{i-j} (w_{i+j}) is 1 if the word v_i occurs in context c at the left (right) of the word w at distance j or 0 otherwise ;
- $\vec{c} = (w_1, \cdots, w_{|W|})$ where w_i is 1 if the word v_i occurs in context c, or 0 otherwise, where v_i is a word from the entire text of $\mid W \mid$ words. In this last example the features are all the words in the contexts.

The similarity between two contexts c_a, c_b (of the same word or different words) is the *normalised cosine* between the vectors $\vec{c_a}$ and $\vec{c_b}$ [7]:

$$cos(\vec{c_a}, \vec{c_b}) = \frac{\sum_{j=1}^{m} w_{a,j} \times w_{b,j}}{\sqrt{\sum_{j=1}^{m} w_{a,j}^2 \times \sum_{j=1}^{m} w_{b,j}^2}}$$

and $sim(\vec{c_a}, \vec{c_b}) = cos(\vec{c_a}, \vec{c_b})$.

In all above examples the number w_i is the weight of th feature v_i. This can be the frequency f_i of the feature v_i (term frequency or tf). On the base of feature

relevance principle, the features can be weighted to reflect the distance of the words to the focus word. For example, in a -3 +3 windows the weights for the 6 features could be: 0.25, 0.5, 1, 1, 0.5, 0.25.

Another method to establish the weight w_i is to capture the fashion of distribution of v_i in all the set of contexts by principle: features that are limited to a small number of contexts are useful for discriminating those contexts; features that occur frequently across the entire set of contexts are less useful in this discrimination. In this case one use a new weight for a feature, called "inverse document frequency", denoted by *idf* and defined as below:

Definition

Let us consider that the number of contexts is Nc and the number of contexts in which the feature v_i occurs is n_i. The *inverse document frequency* is :

$$idf_i = \frac{Nc}{n_i} \; or \; idf_i = log(\frac{Nc}{n_i})$$

Combining the tf with *idf* we obtain $tf.idf$ weighting. In this case: $\vec{c} = (w_1, \cdots, w_{Nf})$, where $w_i = f_i \times idf_i$.

2.3.1. *Second-order co-occurrence.* In [14] the author introduces two types of vectors: word vectors and context vectors. The word vector for a word x is $\vec{x} = (w_1, \cdots, w_{Nf})$ where w_i is the number of times the word v_i co-occurs in the entire corpus. The features v_i can be selected as above. The context vector for a context of an ambiguous word is obtained by summing the vectors of all the vectors of the words in context. Therefore two contexts are similar if the words in these contexts occur with similar words (or, the contextual representation is similar). This is known as *strong contextual hypothesis*. Second order co-occurrence method is more robust than first-order method (as above).

3. Supervised learning of WSD

In such case a system is presented with a training set consisting of a set of input contexts labeled with their appropriate sense (disambiguated corpus). The task is to build a classifier which correctly classifies new cases based on their context of use. The two most known supervised algorithms are Bayesian classification and K-NN classification.

3.1. **Naive Bayes classifier approach of WSD.** This method was been introduced by gale, 1992. In this frame the context of a word w is treated as a bag of words without structure. What we want to find is the best sense s' for an input

context c_{new} of an ambiguous word w. This is obtained as:

$$s' = argmax_{s_k} P(s_k \mid c_{new}) = argmax_{s_k} \frac{P(c_{new} \mid s_k) \times P(s_k)}{P(c_{new})} =$$

$$= argmax_{s_k} P(c_{new} \mid s_k) \times P(s_k)$$

The independence assumption (naive Bayes assumption) is:

$$P(c_{new} \mid s_k) = P(\{v_i \mid v_i \in c_{new}\} \mid s_k) = \prod_{v_i \in c_{new}} P(v_i \mid s_k)$$

This assumption (often referred to as a *bag of words* model)has two consequences:

- the structure and order of words in context is ignored;
- the presence of one word in the context doesn't depends on the presence of another.

This is clearly not true, but there is a large number of cases in which the algorithm works well.

Finally, $s' = argmax_{s_k} P(s_k) \times \prod_{v_i \in c_{new}} P(v_i \mid s_k)$.

Thus the supervised algorithm is:

- TRAINING Calculate:

$$P(s_k) = \frac{C(s_k)}{nr.of contexts}; P(v_i \mid s_k) = \frac{C(v_i, s_k)}{C(s_k)}$$

- TEST Calculate for a new context c_{new} the appropriate sense:

$$s' = argmax_{s_k} P(s_k \mid c_{new}) = argmax_{s_k} P(s_k) \times \prod_{v_i \in c_{new}} P(v_i \mid s_k).$$

3.2. **k-NN or memory based learning.** At training time, a k-NN model memorizes all the contexts in the training set by their associated features. Later, when proceeds a new context c_{new}, the classifier first selects k contexts in the training set that are closest to c_{new}, then picks a sense for c_{new}.

This supervised algorithm is:

- TRAINING Calculate $\vec{c}$ for each context c.
- TEST Calculate:

$$A = \{\vec{c} \mid sim(\vec{c_{new}}, \vec{c}) \; is \; maxim, \mid A \mid = k\}$$

that means A is the set of the k nearest neighbors contexts of $\vec{c_{new}}$.

$$Score(c_{new}, s_j) = \sum_{c_i \in A} (sim(\vec{c_{new}}, \vec{c_i}) \times a_{ij})$$

where a_{ij} is 1 if $\vec{c_i}$ has the sense s_j and a_{ij} is 0 otherwise.
Finally, $s' = argmax_j Score(c_{new}, s_j)$.

3.3. **Bootstrapping approach of WSD.** A major problem with supervised approaches is the need for a large sense tagged training set. The bootstrapping methods use a small number of contexts labeled with senses having a high degree of confidence. This could be accomplished by hand tagging with senses the contexts of an ambiguous word w for which the sense of w is clear because some *seed collocations* [19] occur in these contexts.

These labeled contexts are used as seeds to train an initial classifier. This is then used to extract a larger training set from the remaining untagged contexts. Repeating this process the number of training contexts grows and the number of untagged contexts reduces. We will stop when the remaining unannotated corpus is empty or any new context can't be annotated.

The bootstrapping approach is situated between the unsupervised and unsupervised approach of WSD.

For the word *bass* for example, we might begin with $fish$ as a resonable sense for sense $bass^1$ (bass as fish), as presented in WordNet [4] and *play* as a reasonable sense for $bass^2$ (bass as music). A small number of contexts can be labeled with the sense 1 and 2. These labeled contexts are used to extract a larger set of labeled contexts.

In [16] we present an original algorithm which combines this bootstrapping idea with elements of NB algorithms .

4. Unsupervised approach

Unsupervised approach of WSD does not use sense tagged data (training data) at all. Strictly speaking, the task of unsupervised disambiguations is of *sense discrimination* . In this case, vector representations of unlabeled contexts are grouped into clusters, according to a similarity measure. One cluster is considered as representing a sense and a new context c_{new} is classified as having the sense of the cluster to which it is closest according to the similarity measure. An advantage of unsupervised methods in disambiguation is that granularity of sense distinction is an adjustable parameter: a number of 10 clusters induces more fine-grained sense distinction than a number of 2 clusters, for example.

Let us consider that the objects to be clusterized are the vectors of n words, $\{w_1, w_2, \cdots, w_n\}$. A vector

$$\vec{w_i} = (w_i^1, w_i^2, \cdots, w_i^m)$$

is associated with a word w_i as above.

As clustering methods we can use an agglomerative or divisive hierarchical algorithm or a non-hierarchical (flat) clustering algorithm [16, 1]. In the first case each of the unlabeled context is initially assigned to its own cluster. New clusters are then formed in bottom-up fashion by successively fusion of two clusters that are most similar. This process continues until either a specified number of clusters is obtained or some condition about similarity measure between the clusters is accomplished. In generally, a good clustering method is defined as one that maximizes the within cluster similarity and minimizes the between cluster similarity.

- Agglomerative algorithm for hierarchical clustering [9]. The clustering algorithm begins by considering each word in its own cluster and ends when all the words are in the same cluster.
- Non-hierarchical clustering algorithm [9]. A non-hierarchical algorithm starts out with a partition based on randomly selected seeds (one seed per cluster), and then refine this initial partition. The algorithm stops when a measure of cluster quality is accomplished. As such measure we could select: group average similarity (average similarity between members); single link similarity (the similarity of two most similar elements of a cluster); complete-link similarity (the similarity of two least similar elements from a cluster).

  One of the non-hierarchical algorithm is **k-means**; it defines clusters as the mean (the average) of their member.

One other algorithm in unsupervised approach is EM-algorithm. In this case we start with a random computing of parameters $P(v_j \mid s_k)$ and then this parameters are reestimated in an estimation-modification cycle.

5. Dictionary-based disambiguation.

Work in WSD reached a turning point in the 1980s when large-scale lexical resources such as dictionaries, became widely available. The machine readable dictionaries (MRD) have a large development in these days. This section describes disambiguation methods that rely on the definition of senses of a word in dictionaries and thesauri.

5.1. **Lesk's algorithm. Reduced form**

Lesk (1986) starts from the idea that a word's dictionary definition is a good indicator for the senses of this word. He uses the definition in the dictionary directly.

Suppose that for a polysemic word w we have in a dictionary Ns senses $s_1, s_2, \cdots, s_{Ns}$ given an equal number of definitions $D_1, D_2, \cdots, D_{Ns}$. The new context to be disambiguated is c_{new}.

The idea of Lesk's algorithm is :

$FOR\ k = 1, \cdots, Ns\ DO$
$\quad score(s_k) =\mid D_k \cap (\cup_{v_j \in c_{new}} \{v_j\}) \mid$
$ENDFOR$
$Calculate\ s' = argmax_k score(s_k)$

The score of a sense is number of words that are shared by the sense definition and context.

The method achieved 50-70% correct disambiguation [9].

5.2. **Two claim about senses: one sense per discourse (OSPD), one sense per collocation (OSPC).** In [19] Yarowsky observes that the sense of a target word is highly consistent within any given document or discourse. This is the content of OSPD principle. For example, if a document is about biological life, then each occurrence of the ambiguous word *plant* is more probably linked with the sense of "living being". If the document is about industrial aspects, then *plant* is more probably linked with the sense *factory*. Of course, the definition of discourse is central to the test of OSPD principle.

On the other hand, the sense of a target word is strongly correlated with certain other words in the same phrasal unit, named collocational features. By a collocation we mean usually first /second /third word to the left /right of the target word. In fact, there are words which collocate with the target word w with a high probability. Such a words are considered as strongest in the disambiguation process (OSPC principle). The algorithm proposed by Yarowsky combines both constraints [9].

6. Recent developments.

6.1. **Evaluation of WSD task.** Given the variety in the studies it is very difficult to compare one method with another. Evaluation of WSD programs has excited a great deal of interest. Producing a gold standard corpus annotated corpus is both expansive (many person-months of annotator effort) and hard (different individuals will often assign different senses to the same word-in-context). In

April 1997 a workshop of ACL included first time a session of WSD evaluation [12]. Beginning with 1998 (then in Sussex, England) in each two years take place some WSD evaluation workshops, named SENSEVAL. If at the first edition participated over 20 systems and most research has been in English, in 2004 for the first time was a section for Romanian language where participated 7 teams. For Romanian the manually sense-tagged was worked on a site open at University of North Texas. Almost half the systems used supervised training methods. The evaluation involves comparison of the output of each system using as measures *precision* and *recall*.

The upper bound for accuracy of a WSD system is usually human performance. This is between 97% and 99 % [9]. The lower bound is the performance of the simplest algorithm, baseline, usually the assignment of all contexts to the most frequent sense.

6.2. **Disambiguation and Information Retrieval.** WSD is only an intermediate task in NLP, like POS tagging or parsing. Examples of final applications for which WSD is potentially an issue are: Machine Translation, Information Retrieval (IR), Dialogue systems or improving Parsing. For example, the problem of finding whether a particular sense is connected with an instance of a word is likely the IR task of finding whether a document is relevant to a query. It is established that a good WSD program can improve performance of retrieval by 2%. As IR is used by millions of users, an average of 4 % improvement could be seen as very significant. A test in 1993 compared two term-expansion queries methods for IR: one in which each term was expanded with all related terms and one in which it was only expanded with terms related to the sense used in the query (disambiguated). The conclusion was that disambiguation did not improve the performance of term expansion. In [14] the authors propose a new methods that is beneficial for IR. In this method the features in the definition of vectors are senses, and not words: a feature in a context has a nonzero value if the context contains a word assigned to the sense represented by the feature. This method increased performance by 7,4 % compared to "features equal words" case. The two methods are opposites of each other in the following sense. Term expansion by related terms increases the number of matching documents for a query: if the query contains the word *cosmonaut* and expansion adds *astronaut*, then the number of documents is bigger (the documents containing the word *astronaut* are added). If the word *suit* occurs in the query used in the "legal" sense, then documents that contain *suit*, for example, in the "clothes" sense will not longer be founded. An excellent overview of work in WSD and IR can be found in [6].

References

[1] Avram, Lupşa, D., Şerban, G., Tătar, D.: Hierarhical clustering algorithms for repeating similarity values, Studia Universitatis "Babes-Bolyai", seria Informatica, 2003, nr 2, pp 61 - 72.

[2] Dagan, I., Lee, L., Pereira, F.: Similarity-based models of Word Cooccurences Probabilities, *MLJ*, 34(1-3), 1999.

[3] Daelemans, W.: Machine learning approach in *Syntactic Wordclass Tagging*, Kluwer Academic Publishers, pp 285-304, 1999.

[4] Fellbaum, C. (editor): *WordNet An Electronic Lexical Database*, The MIT Press, 1998.

[5] Gauch, S., Wang, J., Rachakonda, S. M.: A corpus analysis approach for automatic query expansion and its extension to multiple databases, *CIKM'97- Information and Knowledge management*.

[6] Ide, N., Veronis, J.: Introduction to the special issue on WSD: the state of the art, *Computational Linguistics*, 24(1) 1998, pp1-40.

[7] Jurafsky, D., Martin, J.: *Speech and language processing*, Prentice Hall, 2000.

[8] Kilgarriff, A.: *What is WSD good for?*, ITRI Technical Report Series- August, 1997.

[9] Manning, C., Schutze, H. : *Foundation of statistical natural language processing*, MIT, 1999.

[10] Marcus, S. : *Lingvistică matematică*, Ed. Didactică si Pedagogică, Bucureşti, 1966.

[11] Lin, D.:Automatic retrieval and clustering of similar words, *COLING-ACL'98*, Montreal, 1998.

[12] Resnik, P., Yarowsky,D. : Distinguishing Systems and Distinguishing sense: new evaluation methods for WSD , *Natural Language Engineering*, 1 , nr 1, 1998.

[13] Sahlgren, M. : Vector-based semantic analysis: representing word meanings based on random labels, in *The Acquisition and Representation of Word Meaning*,Kluwer Academic Publishers, 2001.

[14] Schutze, H.: Automatic Word Sense Discrimination, Computational Linguistics, *Computational Linguistics*, 24(1) 1998, pp97-123.

[15] Serban, G., Tatar, D.: UBB system at Senseval3, *Proceedings of Workshop in Word Disambiguation*, ACL 2004, Barcelona , July 2004 , pp 226-229.

[16] Tatar, D., Serban,G.: A new algorithm for WSD, *Studia Univ. "Babes-Bolyai", Informatica*, 2001, nr.2, pp 99-108.

[17] Tatar, D.: Inteligenta artificiala: demonstrare automata de teoreme, prelucrarea limbajului natural, *Editura Albastra, Microinformatica*, 2001.

[18] Widdows, D.: A mathematical model for context and word meaning, *Fourth International Conference on Modeling and using context*, Stanford, California, June 23-25, 2003.

[19] Yarowsky, D.: *Hierarchical Decision Lists for WSD*, Kluwer Acadmic Publishers, 1999.

Department of Computer Science, Babeş-Bolyai University, Cluj-Napoca
E-mail address: dtatar@cs.ubbcluj.ro