Fundamenta Informaticae 91 (2009) 547–562
DOI 10.3233/FI-2009-0057
IOS Press

Adjective Sense Disambiguation at the Border Between Unsupervised and Knowledge-Based Techniques

Florentina Hristea*
Department of Computer Science
Faculty of Mathematics and Computer Science, University of Bucharest, Romania
fhristea@fmi.unibuc.ro

Marius Popescu
Department of Computer Science
Faculty of Mathematics and Computer Science, University of Bucharest, Romania
mpopescu@phobos.cs.unibuc.ro

Abstract. The present paper extends a new word sense disambiguation method [9] to the case of adjectives. The method lies at the border between unsupervised and knowledge-based techniques. It performs unsupervised word sense disambiguation based on an underlying Naïve Bayes model, while using WordNet as knowledge source for feature selection. The proposed extension of the disambiguation method makes ample use of the WordNet semantic relations that are typical of adjectives. Its performance is compared to that of previous approaches that rely on completely different feature sets. Test results show that feature selection using a knowledge source of type WordNet is more effective in the disambiguation of adjective senses than local type features (like part-of-speech tags) are.

Keywords: word sense disambiguation, unsupervised disambiguation, knowledge-based disambiguation, Bayesian classification, the EM algorithm

*Address for correspondence: Department of Computer Science, Faculty of Mathematics and Computer Science, University of Bucharest, Romania

1. Introduction

Word sense disambiguation (WSD), which signifies determining the meaning of a word in a specific context, is a core research problem in computational linguistics and natural language processing, which was recognized since the beginning of the scientific interest in machine translation, and in artificial intelligence, in general. Finding a solution to the WSD problem is obviously essential for applications which deal with natural language understanding (message understanding, man-machine communication etc.) and is at least useful, and in some cases compulsory, for applications which do not have natural language understanding as main goal, applications such as: information retrieval, machine translation, speech processing, text processing etc.

As a computational problem lexical disambiguation was often described as being AI-complete. This view originated in the fact that possible statistical approaches to the problem were almost completely ignored in the past. As it is well known, starting with the early nineties, the AI community witnesses a great revival of empirical methods, especially statistical ones. This is due to the success of statistical approaches, as well as of machine learning, in solving problems such as speech recognition or part-of-speech tagging. Nowadays statistical methods and machine learning algorithms are used for solving a great number of problems posed by artificial intelligence, in general, and by natural language processing, in particular. In the subfield of natural language processing (from the perspective of which we shall approach WSD within the framework of the present paper), the problem we are discussing here is defined as that of computationally determining which sense of a word is activated by the use of that word in a particular context and represents, essentially, a classification problem.

The importance of WSD has been widely acknowledged in recent years, with some 700 scientific papers in the ACL Anthology mentioning the term ”word sense disambiguation” and with three classes of WSD methods being taken into consideration by the literature: supervised disambiguation, unsupervised disambiguation and knowledge-based disambiguation.

The supervised approach to WSD consists of automatically inducing classification models or rules from annotated examples. A disambiguated corpus is available for training. This disambiguated corpus will be used in training a classifier that can label words within a new, unannotated text. The task is that of conceiving a classifier which correctly classifies the new cases, based on the context where they occur. One such classifier, that has been widely used in supervised disambiguation, is the Bayes classifier, which builds a probabilistic model while looking at the words around an ambiguous word in a so-called context window.

Unlike supervised disambiguation, the unsupervised approach to the same problem uses no pre-existing knowledge source. Unsupervised disambiguation methods are data-driven, highly portable, robust, and offer the advantage of being language-independent. They rely either on the distributional characteristics of unannotated corpora (which will represent the approach within the present paper), or on translational equivalences in word aligned parallel text. Within the framework of the present study, the term ”unsupervised” will refer, as in [17], to knowledge-lean methods, that do not rely on external knowledge sources such as machine readable dictionaries, concept hierarchies, or sense-tagged text. Due to the lack of knowledge they are confronted with, these methods do not assign meanings to words, relative to a pre-existing sense inventory, but rather make distinctions in meaning based on distributional similarity. While not performing a straightforward WSD, these methods achieve a discrimination among the meanings of a polysemous word. They have the potential to overcome the knowledge acquisition bottleneck (manual sense-tagging). Unsupervised disambiguation also offers the advantage that it can

be easily adapted to produce distinctions between usage types that are more fine grained than would be found in a dictionary. (For example, it can distinguish between "civil suit" and "criminal suit", while regular dictionaries record only "law suit"[1]). Information retrieval is an application for which this is extremely useful.

Finally, knowledge-based disambiguation methods perform sense disambiguation (and not sense discrimination) by means of a pre-existing sense inventory and can be applied to all words of a given text (unlike the techniques based on corpora, which can be used only in the case of those words for which annotated corpora are available).

With the exception of the case when it is unsupervised, the problem of WSD requires establishing a sense inventory, namely determining all meanings which can be assigned to each word that must be disambiguated. While an official and unique sense inventory for English still doesn't exist, Princeton University's WordNet (WN) [13], [14], [15], [7] has probably become the most widely used source for establishing a sense inventory.

WSD can be performed at many levels of granularity. The various existing sense inventories have different such levels of granularity, with WN being very fine-grained. The level of granularity offered by the sense inventory has great influence over WSD, making the problem more or less difficult, and is therefore taken into account in the evaluation of WSD systems.

The study described in the present paper will make major use of WordNet when determining the features necessary for performing unsupervised word sense disambiguation with an underlying Naïve Bayes model. As a result of using the semantic network WordNet, the disambiguation process will be regarded as taking place at the border between unsupervised and knowledge based techniques.

The present paper concentrates on distributional approaches to unsupervised word sense disambiguation that rely on monolingual corpora, with focus on the usage of the Naïve Bayes model in unsupervised WSD. We are given I sentences that each contain a particular polysemous word. Our objective is to divide these I instances of the ambiguous word (the so-called target word) into a specified number of sense groups. These sense groups must be mapped to sense tags in order to evaluate system performance. Let us note that sense tags will be used only in the evaluation of the sense groups found by the unsupervised learning procedure. The discussed algorithm is automatic and unsupervised in both training and application.

From the wide range of unsupervised learning techniques that could be applied to our problem, we have chosen to use a parametric model in order to assign a sense group to each ambiguous occurrence of the target word. In each case, we shall assign the most probable group given the context as defined by the Naïve Bayes model, where the parameter estimates are formulated via unsupervised techniques.

The paper focuses on polysemous English adjectives with the aim of performing unsupervised adjective sense disambiguation with an underlying Naïve Bayes model. It makes use of a new disambiguation method that has been introduced in [9] and tested only with regard to nouns (which have so far attained the highest accuracy with respect to WSD). The paper extends the mentioned method for usage in the case of adjectives and demonstrates that disambiguation results can improve corresponding to this part of speech.

The theoretical model will be presented and its implementation will be discussed. Special attention will be paid to feature selection and parameter estimation, the two main issues of the model's implementation.

[1] Example from [13, p.255].

Unlike previous approaches [19] that, when implementing the same model, make use of a small number of local features which include co-occurrence and part of speech information near the target word, the method we are extending here means to implement a Naïve Bayes model that uses as features the actual words occurring in the context window of the ambiguous word. Our study implements the model (corresponding to the adjective case) in its simplest and most straightforward form, an attempt which, to our knowledge, has not been reported in the literature so far. In order to decrease the number of features (words) used and, as a result, to increase the performance of the disambiguation process [9], our method selects a restricted number of features. Feature selection is performed entirely by using the WordNet semantic network, a choice which places the disambiguation process at the border between unsupervised and knowledge based techniques. Specifically, using as features the content words of the WordNet glosses (together with other words indicated as being relevant by the WordNet semantic relations) places our disambiguation method close to the classical Lesk algorithm [11], in this case the overlap measure being a probabilistic one.

We consider adapting the newly proposed method to the adjective case a topic of genuine interest since adjectives have a completely different organization from that of nouns in WordNet. (Adjective synsets are regarded as clusters of adjectives while noun synsets are part of noun hierarchies). Consequently, semantic relations used in performing feature selection may differ, in accordance with the involved part of speech. In fact, WordNet provides various semantic relations that are typical of specific parts of speech (as is the case of the similarity relation that only holds for adjective synsets contained in adjective clusters).

As a result of the performed adaptation, the obtained adjective sense disambiguation method and corresponding results, as compared to previously existing ones, will reinforce the benefits of combining the unsupervised approach to the WSD problem with a knowledge source of type WordNet.

2. Unsupervised Word Sense Disambiguation with an Underlying Naïve Bayes Model

The algorithm for word sense disambiguation that is studied here exemplifies an important theoretical approach in statistical language processing: Bayesian classification [8]. The idea of the Bayes classifier is that it looks at the words around an ambiguous word in a large context window. Each content word contributes potentially useful information about which sense of the ambiguous word is likely to be used with it. The classifier does no feature selection. Instead it combines the evidence from all features. The mentioned classifier [8] is an instance of a particular kind of Bayes classifier, the Naïve Bayes classifier.

As commented in [12], Naïve Bayes is widely used due to its efficiency and its ability to combine evidence from a large number of features. It is applicable if the state of the world that we base our classification on is described as a series of attributes. In our case, we describe the context of the ambiguous word in terms of the words that occur in the context. The Naïve Bayes assumption is that the attributes used for description are all conditionally independent. Again as noted in [12], in this case, the Naïve Bayes assumption has two consequences. The first is that all the structure and linear ordering of words within the context are ignored, leading to a so-called ”bag of words model”. The other is that the presence of one word in the bag is independent of another, which is clearly not true in the case of natural language. However, in spite of the mentioned simplifying assumption, this model has been proven to be quite effective when put into practice.

2.1. The Model

In order to formalize the described model, we shall be using the following notations:

$\mathcal{C}$ - corpus representing the available data,
w - the word to be disambiguated (target word),
$s_1, \ldots, s_K$ - possible senses of w,
$c_1, \ldots, c_I$ - contexts of w in a corpus,
$v_1, \ldots, v_J$ - words used as contextual features for the disambiguation of w,
$C(v_j, c_i)$ - the number of occurrences of word v_j in context c_i,
$P(s_k) = \alpha_k$, $k = \overline{1,K}$,
$P(v_j|s_k) = \theta_{kj}$, $k = \overline{1,K}$; $j = \overline{1,J}$
h_{ik} - senses of the ambiguous words, given by

$$h_{ik} = \begin{cases} 1, & \text{context } c_i \text{ generates sense } s_k,\ i = \overline{1,I};\ k = \overline{1,K} \\ 0, & \text{otherwise} \end{cases},$$

Ψ - the parameter vector (used by the EM algorithm).

The contextual features $v_1, \ldots, v_J$ occur in a fixed position near w, in a window of fixed length, centered or not on w[2]. A Naïve Bayes classifier computes the sense s', which for the target word w and a given context c, satisfies the relation

$$s' = \underset{k=\overline{1,K}}{\operatorname{argmax}} P(s_k|c). \tag{1}$$

Sense s' will represent the result of our disambiguation process. The value of $P(s_k|c)$ in (1) can be computed using Bayes' rule:

$$P(s_k|c) = \frac{P(s_k)P(c|s_k)}{\sum_{k=1}^{K} P(s_k) \cdot P(c|s_k)}. \tag{2}$$

When neglecting the denominator in (2) and using logs of probabilities to make the computation simpler, formula (1) becomes:

$$s' = \underset{k=\overline{1,K}}{\operatorname{argmax}} [\log P(s_k) + \log P(c|s_k)]. \tag{3}$$

The values $[\log P(s_k) + \log P(c|s_k)]$, $k = \overline{1,K}$, must be estimated. The available data is represented by a corpus $\mathcal{C}$, in which I independent contexts $c_i (i = \overline{1,I})$ of the target word w occur. The likelihood of the corpus $\mathcal{C}$ is the product of the probabilities $P(c_i)$ of the individual contexts c_i, and is therefore given by:

$$P(\mathcal{C}) = \prod_{i=1}^{I} P(c_i) = \prod_{i=1}^{I} \sum_{k=1}^{K} P(s_k)P(c_i|s_k). \tag{4}$$

[2]In what follows, a window of size n will denote taking into consideration n content words to the left and n content words to the right of the target word, whenever possible. The total number of words taken into consideration for disambiguation will therefore be $2n + 1$. When not enough words are available, the entire sentence in which the target word occurs will represent the window of context.

In order to compute $P(c_i|s_k), i = \overline{1,I}; k = \overline{1,K}$, we use a Naïve Bayes model with J independent attributes:

$$P(c_i|s_k) = \prod_{v_j \text{ in } c_i} P(v_j|s_k) = \prod_{j=1}^{J} (P(v_j|s_k))^{C(v_j,c_i)}, \tag{5}$$

The likelihood of the corpus C then becomes:

$$P(\mathcal{C}) = \prod_{i=1}^{I} \sum_{k=1}^{K} P(s_k) \prod_{j=1}^{J} (P(v_j|s_k))^{C(v_j,c_i)}. \tag{6}$$

In the Naïve Bayes model, all features are assumed to be conditionally independent given the value of the classification variable. When applied to word sense disambiguation, the model specifies that all contextual features are conditionally independent given the sense of the ambiguous word. The parameters of the above described model are $P(s_k)$ and $P(v_j|s_k)$, denoted α_k and θ_{kj}, respectively.

In supervised disambiguation θ_{kj} and α_k are computed via Maximum - Likelihood estimation, perhaps with appropriate smoothing, from the labeled training corpus. Parameter estimation in unsupervised disambiguation, however, is not based on a labeled training set. Instead, we start with a random initialization of the parameters. The parameters are then reestimated by the EM algorithm so as to maximize the likelihood of the data given the model. Concerning α_k and θ_{kj}, the following conditions must be met:

$$\sum_{k=1}^{K} \alpha_k = 1, \ \sum_{j=1}^{J} \theta_{kj} = 1, \ k = \overline{1,K}. \tag{7}$$

The likelihood of the corpus $\mathcal{C}$ now becomes:

$$P(\mathcal{C}) = \prod_{i=1}^{I} \sum_{k=1}^{K} \alpha_k \prod_{j=1}^{J} \theta_{kj}^{C(v_j,c_i)}. \tag{8}$$

The parameters α_k and θ_{kj} will be estimated by means of the EM algorithm.

2.2. Parameter Estimation by Means of the EM Algorithm

As it is well known, when the likelihood function is not well approximated by a normal distribution, simulation techniques often provide better estimates of the model parameters. For the presented model we shall employ the Expectation Maximization (EM) algorithm [6] in order to estimate model parameters from untagged data. The EM algorithm is a widely used iterative algorithm that formulates a maximum likelihood estimate of each model parameter in the presence of missing data. In our case, the missing data are the senses of the ambiguous words.

There are two steps in the EM algorithm [6], expectation (E-step) and maximization (M-step). The E-step calculates the expected values of the sufficient statistics given the current parameter estimates. The M-step makes maximum likelihood estimates of the parameters given the imputed values of the sufficient statistics. These steps alternate until the parameter estimates in iteration $r-1$ and r differ by less than ε.

In the case of the previously described model, the observed data are given by corpus C, while the missing data are the senses of the ambiguous words, given by h_{ik}.

Each M-step of the algorithm computes the maximum likelihood estimate corresponding to the likelihood of the complete data (which consists of the observed data and the missing data).

The likelihood of the complete data is given by

$$P_{comp}(\mathcal{C}) = \prod_{i=1}^{I}\prod_{k=1}^{K}\Big(\alpha_k \prod_{j=1}^{J}\theta_{kj}^{C(v_j,c_i)}\Big)^{h_{ik}}. \tag{9}$$

We therefore have

$$\log\ P_{comp}(\mathcal{C}) = \sum_{i=1}^{I}\sum_{k=1}^{K} h_{ik}\Big(\ln\alpha_k + \sum_{j=1}^{J} C(v_j,c_i)\ln\theta_{kj}\Big). \tag{10}$$

The problem to be solved is given by the following system:

$$\begin{cases} \max\ \ln P_{comp}(\mathrm{C}) \\ \sum\limits_{k=1}^{K}\alpha_k = 1 \\ \sum\limits_{j=1}^{J}\theta_{kj} = 1,\ k=\overline{1,K} \end{cases}. \tag{11}$$

Parameter estimation via the EM algorithm starts with a random initialization of the parameters. Let $\Psi^{(0)}$ represent the initial value of the parameter vector Ψ,

$$\Psi^{(0)} = \Big(\alpha_1^{(0)},\ldots,\alpha_k^{(0)},\theta_{11}^{(0)},\ldots,\theta_{KJ}^{(0)}\Big), \tag{12}$$

and let us take into consideration *iteration* $(r+1)$ of the EM algorithm. Then the *E-step* computes

$$h_{ik}^{(r)} = P_{\Psi^{(r)}}(h_{ik}=1|\mathcal{C}), \tag{13}$$

the probability of sense s_k generating context c_i when using the model parameters estimated at iteration r, as follows:

$$h_{ik}^{(r)} = \frac{\alpha_k^{(r)}\cdot\prod\limits_{j=1}^{J}(\theta_{kj}^{(r)})^{C(v_j,c_i)}}{\sum\limits_{k=1}^{K}\alpha_k^{(r)}\prod\limits_{j=1}^{J}(\theta_{kj}^{(r)})^{C(v_j,c_i)}}. \tag{14}$$

The *M-step* computes $\alpha_k^{(r+1)}$ and $\theta_{kj}^{(r+1)}$ as follows:

$$\alpha_k^{(r+1)} = \frac{1}{I}\sum_{i=1}^{I} h_{ik}^{(r)},\ k=\overline{1,K} \tag{15}$$

$$\theta_{kj}^{(r+1)} = \frac{\sum_{i=1}^{I} C(v_j, c_i) \cdot h_{ik}^{(r)}}{\sum_{j=1}^{J} \sum_{i=1}^{I} C(v_j, c_i) \cdot h_{ik}^{(r)}}, \quad k = \overline{1, K}. \tag{16}$$

The stopping criterion for the algorithm is given by

$$||\Psi^{(r+1)} - \Psi^{(r)}||^2 < \epsilon, \tag{17}$$

namely

$$\sum_{k=1}^{K} \left(\alpha_k^{(r+1)} - \alpha_k^{(r)}\right)^2 + \sum_{k=1}^{K} \sum_{j=1}^{J} \left(\theta_{kj}^{(r+1)} - \theta_{kj}^{(r)}\right)^2 < \epsilon. \tag{18}$$

The EM algorithm is guaranteed to increase the log likelihood of the data given the model in each step. Therefore, the stopping criterion for the algorithm is to stop when the likelihood is no longer increasing significantly.

Once the parameters of the model have been estimated, we can disambiguate contexts of w by computing the probability of each of the senses based on the words v_j occurring in the context. Making the Naïve Bayes assumption and using the Bayes decision rule, we can decide s' if

$$s' = \underset{s_k}{\operatorname{argmax}} \left[\log P(s_k) + \sum_{v_j \text{ in } c_i} \log P(v_j | s_k) \right]. \tag{19}$$

Our choice of recommending usage of the EM algorithm for parameter estimation in the case of unsupervised disambiguation is based on the fact that this algorithm is known as a very successful iterative method that fits well to models with missing data.

3. Making Use of WordNet for Feature Selection

When the Naïve Bayes model is applied to supervised disambiguation, the actual words occurring in the context window are usually used as features. This type of approach generates a great number of features and, implicitly, a great number of parameters. This can dramatically decrease the model's performance, since the available data is usually insufficient for the estimation of the great number of resulting parameters, a situation which becomes even more drastic in the case of unsupervised disambiguation, where parameters must be estimated in the presence of missing data (the sense labels).

In order to overcome this problem, the various existing unsupervised approaches to WSD implicitly or explicitly perform a feature selection. One could say, in fact, that, when implementing the previously described (§2) model, discussion among specialists focuses almost entirely on the issue of feature selection.

Two early approaches to word sense discrimination, Schütze's context group discrimination [20] and Pedersen and Bruce's McQuitty's Similarity Analysis [18], [19], rely on totally different sets of features and, in our view, still represent the main approaches to feature selection.

As commented in [17] Schütze represents contexts in a high dimensional feature space that is created using a separate large corpus (referred to as the training corpus). He selects features based on their frequency counts or log-likelihood ratios in this corpus. This approach adapts LSI/LSA so that it represents entire contexts rather than single word types using second-order co-occurrences[3] of lexical features. While Schütze reduces dimensions by means of LSI/LSA, Pedersen and Bruce define features over a small contextual window (local context) and select them to produce low dimensional event spaces. They make use of a small number of first-order features to create matrices that show the pairwise (dis)similarity between contexts. They rely on local features that include co-occurrence and part of speech information near the target word. Three different feature sets, consisting of various combinations of features of the mentioned types, were defined in [19] for each word and were used to formulate a Naïve Bayes model describing the distribution of sense groups of that word. Unlike Schütze, Pedersen and Bruce select features from the same test data that is being discriminated, which, as noted in [17], is a common practice in clustering in general.

While local-context features had already been used successfully in a variety of supervised approaches to disambiguation [4], [16], Pedersen and Bruce make good use of them in unsupervised word sense disambiguation [19]. They conducted an experimental evaluation relative to the 12-word sense-tagged corpus [4] of Bruce and Wiebe (1994) as well as with the *line* corpus [10]. In the case of adjectives, no feature set resulted in greater accuracy than the most frequent sense. However, as noted in [17], those sense distributions were rather skewed, with most adjectives having a majority sense between 70% and 90%.

The approach to WSD of the present paper relies on a set of features formed by the actual words occurring near the target word (within the context window) and tries to reduce the size of this feature set by performing knowledge-based feature selection. The semantic network WordNet (WN) will be used as unique knowledge source for feature selection. While the classical approach forms the vocabulary on which the disambiguation process relies dynamically, using all the content words which occur in the contexts, the present approach forms the same vocabulary based entirely on WordNet. According to the new disambiguation method we have introduced in [9] and which we are now extending to the adjective case, the WN semantic network will provide the words considered relevant for the set of senses taken into consideration corresponding to the target word.

First of all, words occurring in the same WN synsets as the target word (WN synonyms) have been chosen, corresponding to all senses of the target. Additionally, we have considered as part of the vocabulary used for disambiguation the words occurring in synsets related (through explicit relations provided in WordNet) to those containing the target word. Synsets and relations have been restricted to those associated with the part of speech of the target word. We have equally taken into consideration the content words of the glosses of all types of synsets participating in the disambiguation process, using the example string associated with the synset gloss, as well. The latter choice has been made since previous studies [3], performed for knowledge-based disambiguation, have come to the conclusion that the ”example relation”- which simply returns the example string associated with the input synset - seems to provide useful information in the case of all parts of speech.

Corresponding to the studied part of speech, our disambiguation method has taken into account the *similarity* relation, which is typical of adjectives (and, in fact, only holds for adjective synsets contained

[3]Two instances of an ambiguous word are assigned to the same sense if the words that they co-occur with likewise co-occur with similar words in the training data.

in adjective clusters[4]). The *also-see* relation and the *attribute* relation have also been taken into account since thes e relations are considered most informative and have been found [3] to rank highest among the useful relations for adjectives. The *pertaining-to* relation has also been considered, whenever possible. Finally, the *antonymy* relation has represented a source of "negative information" that has proven itself useful in the disambiguation process. This is in accordance with previous findings of studies performed for knowledge-based disambiguation [2] that consider the antonymy relation a source of negative information allowing a disambiguation algorithm "to identify the sense of a word based on the absence of its antonymous sense in the window of context". Tables 3 and 4 of §4.2 show the obtained disambiguation results when using a "disambiguation vocabulary" in the formation of which all mentioned types of synsets have taken part. Disambiguation results are computed with and without antonym synsets participating in the disambiguation process. As a result of using only those words indicated as being relevant by WordNet, a much smaller vocabulary is obtained, and therefore a much smaller number of features will take part in disambiguation.

As commented in [9], we consider that this manner of performing feature selection (in this case, relative to adjectives) brings our disambiguation method and corresponding algorithm close to what is known in the literature [2] as "the adapted Lesk algorithm", which uses related synsets and the corresponding extended gloss overlaps as compared to the original Lesk algorithm [11].

In the case of our proposed disambiguation method, the features represent the number of occurrences in the given context (window) of a word belonging to the vocabulary. The disambiguation vocabulary created by us, based on WordNet, can be regarded as representing the (extended) sense definitions, in the sense of the adapted Lesk algorithm. The main difference when comparing to the Lesk/adapted Lesk algorithm consists in the way in which a word (feature) contributes to the final score being assigned to a sense. In the case of the Lesk algorithm each word (feature) contributes to the final sense score with the same weight (1), while with adapted Lesk scores are based on the length of a match[5]. In the case of our method each word (feature) contributes to the same score with a weight given by $P(v_j|s_k)$. This weight (probability) is not a priori established, but is learned by means of the EM algorithm.

4. Empirical Evaluation

4.1. Design of the Experiments

Our disambiguation results concerning adjectives will be compared with those of [19] where an algorithm of the same type (unsupervised with an underlying Naïve Bayes Model) is placed under survey. However, the algorithm studied by Pedersen and Bruce relies on a restricted set of local features, that include co-occurrence and part of speech information near the target word (as commented in §3). It therefore also performs feature selection, although in a completely different manner than that proposed in the present paper.

[4]WordNet divides adjectives into two major classes: descriptive and relational. Descriptive adjectives are organized into clusters on the basis of binary opposition (antonymy) and similarity of meaning [7]. Descriptive adjectives that do not have direct antonyms are said to have indirect antonyms by virtue of their semantic similarity to adjectives that do have direct antonyms. Relational adjectives are assumed to be stylistic variants of modifying nouns and are cross-referenced to the noun files (see the relation "relating-or-pertaining-to"). The function such adjectives play is usually that of classifying their head nouns [7].
[5]The adapted Lesk algorithm assigns to a n word overlap the score of n^2.

Table 1. Distribution of Senses of *common*

sense	count
as in the phrase ”common stock”:	84%
belonging to or shared by 2 or more:	8%
happening often; usual:	8%
total count:	1060

Table 2. Distribution of Senses of *public*

sense	count
concerning people in general:	68%
concerning the government and people:	19%
not secret or private:	13%
total count:	715

As test data we have used the (Bruce, Wiebe & Pedersen 1996) data [5] containing twelve words taken from the ACL/DCI Wall Street Journal corpus and tagged with senses from the Longman Dictionary of Contemporary English. We have chosen this data set for our tests concerning adjectives since it has equally been used in the case of the (Pedersen and Bruce 1998) approach to WSD, to which we shall be comparing the results of our own disambiguation method. Test results will be reported in the case of two adjectives, *common* and *public*, the latter being the one corresponding to which Pedersen and Bruce obtain the most modest disambiguation results. The senses of *common* that have been taken into consideration and their frequency distribution are shown in Table 1, while Table 2 provides the same type of information corresponding to the adjective *public*. In these tables *total count* represents the number of occurrences in the corpus of each word, with each of the adjectives being limited to the 3 most frequent senses, while *count* gives the percentage of occurrence corresponding to each of these senses.

In fact, our choice of performing tests in the case of adjectives *common* and *public* has been influenced by the fact that these adjectives are represented in the mentioned corpus by three different senses, while the other two adjectives for which Pedersen and Bruce perform disambiguation tests, *chief* and *last*, have only two senses (in the same corpus). Since unsupervised disambiguation should be able to produce distinctions even between usage types that are more fine grained than would be found in a dictionary, as noted in our Introduction, our choice of testing in the case of those adjectives having the greatest number of senses represented in the corpus becomes a natural one.

In order for our experiments to be conducted, the data set was preprocessed in the usual required way for WSD: the stop words were eliminated, and Porter stemmer was applied to the remaining words.

The overall source for creating the disambiguation vocabulary was WordNet 3.0, which lists 9 different senses corresponding to the adjective *common* and only 2 different senses corresponding to the adjective *public*. Obviously, a sense mapping of the initial (corpus) senses to those of the WN 3.0 database was necessary. According to this mapping, 4 WN synsets took part in the disambiguation vocabulary

corresponding to the adjective *common*, namely the synsets having the IDs 300492677[6], 302152473[7], 301673815[8] and 300970610[9], respectively. Both WN synsets corresponding to the adjective *public* and having the IDs 300493297[10] and 301861205[11], respectively were part of the same vocabulary when performing disambiguation tests relative to this adjective.

Let us once again note that our disambiguation method is an unsupervised one and therefore does not require sense labels. Performing the mentioned sense mapping was necessary solely for establishing the restricted disambiguation vocabulary (relevant words).

Once the subset of WN senses taking part in the experiments has been established, the relevant information for building the vocabulary must be specified.

Each of the experiments involving the disambiguation of adjectives *common* and *public* have established as relevant words forming the vocabulary all words of the WN 3.0 synsets containing the respective adjective which have been chosen as a result of sense mapping. Additionally, all content words occurring in the glosses and the associated example strings of these synsets have been added to this vocabulary. Information provided by the synsets related (through explicit relations existing in WN) to those containing the target word has also been included in the same vocabulary. Thus, the first performed experiment[12] additionally uses all content words occurring in the synsets, their corresponding glosses and example strings, given by the similarity relation, the also-see relation, the attribute relation, the pertaining-to relation, whenever possible, and, finally, the antonymy relation, which has been considered interesti ng due to the ”negative information” it can provide. The second performed experiment[13] eliminates from the disambiguation vocabulary all words brought in precisely by these antonym synsets.

4.2. Test Results

Performance is evaluated in terms of accuracy. In the case of unsupervised disambiguation defining accuracy is not as straightforward as in the supervised case. Our objective is to divide the I given instances of the ambiguous word into a specified number K of sense groups, which are in no way connected to the sense tags existing in the corpus. In our experiments, sense tags are used only in the evaluation of the sense groups found by the unsupervised learning procedure. These sense groups must be mapped to sense tags in order to evaluate system performance. As in previous studies [19] we have used the mapping that results in the highest classification accuracy[14].

[6]This is synset {common} having the gloss ‘belonging to or participated in by a community as a whole; public’.
[7]This is synset {common, mutual} having the gloss ‘common to or shared by two or more parties’.
[8]This is synset {common} having the gloss ‘to be expected; standard’.
[9]This is synset {common, usual} having the gloss ‘commonly encountered’.
[10]This is synset {public} having the gloss ‘affecting the people or community as a whole’.
[11]This is synset {public} having the gloss ‘not private; open to or concerning the people as a whole’.
[12]referred to in Tables 3 and 4 as ”all”.
[13]referred to in Tables 3 and 4 as ”all-antonyms”.
[14]In order to conduct our experiments we have chosen a number of sense groups equal to the number of sense tags existing in the corpus. Therefore a number of $K!$ possible mappings (with K denoting the number of senses of the target word) should be taken into account. For a fixed mapping, its accuracy is given by the number of correct labellings (identical to the corresponding corpus sense tags) divided by the total number of instances. From the $K!$ possible mappings, the one with maximum accuracy has been chosen.

Table 3. Experimental Results for 3 Senses of *common*

Method	No. of features	Percentage of instances having only null features	Accuracy
all	83	19.2	.775±.02
all - antonyms	74	20.0	.766±.04

Table 4. Experimental Results for 3 Senses of *public*

Method	No. of features	Percentage of instances having only null features	Accuracy
all	74	43.3	.559±.03
all - antonyms	71	44.4	.550±.03

Test results are presented in Tables 3 and 4. Each result represents the average accuracy and standard deviation obtained by the learning procedure over 20 random trials while using a context window of size 25[15] and a threshold ε having the value 10^{-9}.

Apart from accuracy, the following type of information is also included in Table 3 (corresponding to adjective *common*) and in Table 4 (corresponding to adjective *public*): number of features resulting in each experiment and percentage of instances having only null features (i.e. containing no relevant information).

As previously mentioned, within the present approach to disambiguation, the value of a feature is given by the number of occurrences of the corresponding word in the given context window. Taking into consideration that the process of feature selection is based on the restriction of the disambiguation vocabulary, one must notice it is possible for certain instances not to contain (in their context window) any of the relevant words forming this vocabulary. Such instances will have null values corresponding to all features. The smaller the number of features used for disambiguation, the more frequently this takes place. These instances do not contribute to the learning process. However, they have been taken into account in the evaluation stage of our experiments.

We have compared our disambiguation results primarily to those of Pedersen and Bruce presented in [19], since both disambiguation methods that have been used rely on an underlying Naïve Bayes model, use the EM algorithm for estimating model parameters[16] in unsupervised WSD and perform feature selection. The main difference between the two approaches consists in the way feature selection is performed. While Pedersen and Bruce, as mentioned before, use local features that include co-occurrence

[15]The choice of this context window size is based on the suggestion of [11] that the quantity of data available to the algorithm is one of the biggest factors to influence the quality of disambiguation. In our case, a larger context window allows the occurrence of a greater number of WN relevant words (with respect to the target), which are the only ones to participate in the creation of the disambiguation vocabulary.

[16]Pedersen and Bruce also make use [19] of Gibbs sampling for parameter estimation, sometimes with better results, but without these results improving significantly.

and part of speech information near the target word, the present approach relies on WordNet and its rich set of semantic relations for performing feature selection. This places the disambiguation process at the border between unsupervised and knowledge-based techniques, but improves disambiguation accuracy consisten tly.

Thus, the way in which our method performs feature selection brings a disambiguation accuracy of .775±.02 in the case of the adjective *common*, while the highest accuracy obtained in [19], corresponding to the same adjective and when estimating model parameters with the EM algorithm as well, is of .543±.09. When leaving out antonym synsets the accuracy obtained by our method decreases to .766±.04, which again represents a value significantly higher than the corresponding one of [19]. In the case of adjective *public* our method attains an accuracy of .559±.03, which decreases to .550±.03 when leaving out antonym synsets, with both values being higher than the corresponding one obtained in [19]: .507±.03. These results clearly show that feature selection using a knowledge source of type WordNet can be more effective in disambiguation than local type features (like part-of-speech tags).

When analyzing the results presented in Tables 3 and 4 one must also notice that accuracy decreases each time the information provided by the antonym synsets is left out of the disambiguation vocabulary. Although there is an obviously restricted number of antonym synsets (see the number of features in the tables) the type of negative information they provide seems to be beneficial to the disambiguation process.

Finally, the fact that, although adjective *public* has only two senses in WN 3.0, discrimination among three different senses was possible, reinforces the idea that unsupervised disambiguation is able to make distinctions between very fine grained usage types, even more fine grained than those present in a knowledge source of type WordNet.

Conclusion

The present paper has concentrated on distributional approaches to unsupervised word sense disambiguation that rely on monolingual corpora, with focus on the usage of the Naïve Bayes model in unsupervised WSD and with special reference to adjectives. The theoretical model was presented and its implementation was discussed. Special attention was paid to feature selection, the main issue of the model's implementation. A new method for performing feature selection that has been proposed in [9] was extended to the adjective case and tested corresponding to polysemous English adjectives. The novelty of our proposed method consists in using the semantic network WordNet as knowledge source for feature selection. Our method makes ample use of the WordNet semantic relations which are typical of adjectives (a part of speech that has a completely different organization in WordNet from that of nouns, corresponding to which the method had been previously tested). Usage of WordNet as knowledge source for feature selection places the disambiguation process at the border between unsupervised and knowledge-based techniques. Test results show that feature selection performed in this manner is more effective in the disambiguation of adjective senses than local type features (like part-of-speech tags) are.

Although not totally knowledge-lean, we hope the presentation of our disambiguation method in the case of adjectives has reinforced the benefits of combining the unsupervised approach to the WSD problem with a knowledge source of type WordNet.

Acknowledgements

The authors express their deepest gratitude to Dr. Ted Pedersen for having provided the data set necessary for performing the presented tests and comparisons.

The present study has been funded by the National University Research Council of Romania (the "Ideas" research programme, PNII - IDEI).

References

[1] Agirre, E., Edmonds, P., Eds.: *Word Sense Disambiguation. Algorithms and Applications*, Springer, The Netherlands, 2006.

[2] Banerjee, S., Pedersen, T.: An adapted Lesk algorithm for word sense disambiguation using WordNet, *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics, February 17-23, Mexico City*, 2002.

[3] Banerjee, S., Pedersen, T.: Extended gloss overlaps as a measure of semantic relatedness, *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, August 9-15, Acapulco, Mexico*, 2003.

[4] Bruce, R., Wiebe, J.: Word sense disambiguation using decomposable models, *Proceedings of the* 32^{nd} *Meeting of the Association for Computational Linguistics, June 27-30, Las Cruces, New Mexico*, 1994.

[5] Bruce, R., Wiebe, J., Pedersen, T.: The measure of a model, *Proceedings of the Conference on Empirical Methods in Natural Language Processing, Philadelphia, PA*, 1996.

[6] Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society B*, **39**(1), 1977, 1–38.

[7] Fellbaum, C., Ed.: *WordNet: an Electronic Lexical Database*, The MIT Press, Cambridge, Mass, 1998.

[8] Gale, W., Church, K., Yarowsky, D.: A method for disambiguating word senses in a large corpus, *Computers and the Humanities*, **26**(5-6), 1992, 415–439.

[9] Hristea, F., Popescu, M., Dumitrescu, M.: Performing word sense disambiguation at the border between unsupervised and knowledge-based techniques, *Computational Intelligence*, submitted.

[10] Leacock, C., Towell, G., Voorhees, E.: Corpus-based statistical sense resolution, *Proceedings of the ARPA Workshop on Human Language Technology, Princeton, New Jersey*, 1993.

[11] Lesk, M.: Automatic sense disambiguation: how to tell a pine cone from an ice cream cone, *Proceedings of the 1986 SIGDOC Conference, New York, Association for Computing Machinery*, 1986.

[12] Manning, C., Schütze, H.: *Foundations of Statistical Natural Language Processing*, The MIT Press, Cambridge, Mass., 1999.

[13] Miller, G.: Nouns in WordNet: a lexical inheritance system, *International Journal of Lexicography*, **3**(4), 1990, 245–264.

[14] Miller, G.: WordNet: a lexical database, *Communications of the ACM*, **38**(11), 1995, 39–41.

[15] Miller, G., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: WordNet: an on-line lexical database, *Journal of Lexicography*, **3**(4), 1990, 234–244.

[16] Ng, H., Lee, H.: Integrating multiple knowledge sources to disambiguate word sense: an exemplar-based approach, *Proceedings of the* 34^{th} *Annual Meeting of the Society for Computational Linguistics, Santa Cruz, California*, 1996.

[17] Pedersen, T.: Unsupervised corpus-based methods for WSD, *Word Sense Disambiguation. Algorithms and Applications. Edited by E. Agirre and P. Edmonds, Springer, the Netherlands*, 2006.

[18] Pedersen, T., Bruce, R.: Distinguishing word senses in untagged text, *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, 1997.

[19] Pedersen, T., Bruce, R.: Knowledge lean word sense disambiguation, *Proceedings of the* 15^{th} *National Conference on Artificial Intelligence, Madison, Wisconsin*, 1998.

[20] Schütze, H.: Automatic word sense discrimination, *Computational Linguistics*, **24**(1), 1998, 97–123.