Artif Intell Rev
DOI 10.1007/s10462-009-9117-6

Performing word sense disambiguation at the border between unsupervised and knowledge-based techniques

Florentina Hristea · Marius Popescu · Monica Dumitrescu

Abstract This paper aims to fully present a new word sense disambiguation method that has been introduced in Hristea and Popescu (Fundam Inform 91(3–4):547–562, 2009) and so far tested in the case of adjectives (Hristea and Popescu in Fundam Inform 91(3–4):547–562, 2009) and verbs (Hristea in Int Rev Comput Softw 4(1):58–67, 2009). We hereby extend the method to the case of nouns and draw conclusions regarding its performance with respect to all these parts of speech. 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 performance of the method is compared to that of previous approaches that rely on completely different feature sets. Test results for all involved parts of speech show that feature selection using a knowledge source of type WordNet is more effective in disambiguation than local type features (like part-of-speech tags) are.

Keywords Word sense disambiguation · Unsupervised disambiguation · Knowledge-based disambiguation · Bayesian classification · The EM algorithm · WordNet

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

F. Hristea (✉) · M. Popescu · M. Dumitrescu
University of Bucharest, Academiei 14, Str., Sector 1, 010014 Bucharest, Romania
e-mail: fhristea@mailbox.ro; fhristea@fmi.unibuc.ro

Published online: 23 September 2009

goal, applications such as: information retrieval, machine translation, speech processing, text processing etc.

In the subfield of natural language processing (from the perspective of which we shall approach WSD within the framework of the present study), 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.

This problem becomes even more important and difficult to solve when taking into account the great existing number of natural languages with very high polisemy. As noted in Agirre and Edmonds (2006), the 121 most frequent English nouns, for instance, which account for about one in five word occurrences in real English text, have on average 7.8 meanings each, according to the Princeton University lexical database WordNet (Miller 1990; Miller et al. 1990).

The importance of WSD has been widely acknowledged in recent years, with some 700 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.

Supervised disambiguation is based on learning. As it is well known, 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 looks 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 Pedersen (2006), 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. As commented in Agirre and Edmonds (2006), they have the potential to overcome the new knowledge acquisition bottleneck (manual sense-tagging).

Finally, knowledge-based disambiguation methods perform sense disambiguation (and not sense discrimination) by means of a pre-existing sense inventory. These methods can usually 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. However, the concept of word sense still generates debates among linguists. That is probably why, nowadays, an official and unique sense inventory for English still doesn't exist. Some of the most frequent sources used for establishing a sense inventory are: electronic dictionaries, thesauri (LDOCE, Roget's Thesaurus), bilingual

dictionaries in electronic format, and lexical knowledge bases (of type WordNet). Princeton University's WordNet[1] (Miller 1990, 1995; Miller et al. 1990; Fellbaum 1998) has probably become the most widely used source for establishing a sense inventory.

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 freely available resource 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, as in previous studies (Pedersen and Bruce 1998), 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 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.

Unlike previous approaches (Pedersen and Bruce 1998) 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 present paper 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 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, a restricted number of features will be selected. Feature selection will be performed entirely by using the WordNet semantic network, a choice which places the disambiguation process at the border between unsupervised and knowledge based techniques. The obtained 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 WordNet

As its authors note, WordNet (WN) is a lexical knowledge base, first developed for English and then adopted for several European languages, which was created as a machine-readable dictionary based on psycholinguistic principles. It is a lexical database that currently contains (ver. 3.0) approximately 155,287 English nouns, verbs, adjectives and adverbs organized by semantic relations into 117,659 meanings, where a meaning is represented by a set of

[1] Available at http://wordnet.princeton.edu/.

synonyms (a synset) that can be used (in an appropriate context) to express that meaning. Different relations link the synonym sets. An entry in WN consists of a synset, a definitional gloss, and (sometimes) one or more phrases illustrating usage. The semantic relations used to organize words and entries are synonymy and antonymy, hyponymy, troponymy and hypernymy, meronymy and holonymy.

WN was primarily viewed as a lexical database. However, due to its structure, it can be equally considered a semantic network and a knowledge base. It has been recognized as a valuable resource in the human language technology and knowledge processing communities. In WSD, WN represents the most popular sense inventory, with its synsets being used as labels for sense disambiguation.

WSD can be performed at many levels of granularity. The various existing sense inventories have different such levels of granularity. WN is very fine-grained, while thesauri and bilingual dictionaries have much lower granularity. 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.

As the American WN continues to grow, new features are added to it. Version 2.1, for instance, is the first to incorporate the distinctions between classes and instances reported in Miller and Hristea (2006) which lead to a semi-ontology of WN nouns. The noun ontology, in fact, represents the best developed portion of WN, and nouns have so far attained the highest accuracy when knowledge-based WSD using the WN sense inventory has been performed. This is the main reason why the present study will concentrate on noun sense disambiguation, although the discussion which is to follow equally applies to all parts of speech (Hristea and Popescu 2009; Hristea 2009).

3 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 (Gale et al. 1992). The idea of the Bayes classifier (in the context of WSD) 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 (Gale et al. 1992) is an instance of a particular kind of Bayes classifier, the Naïve Bayes classifier.

3.1 The probability model of the corpus, the Bayes classifier and parameter estimation

The model we use for the probability structure of corpus $\mathcal{C}$ is the Naïve Bayes Model. The following *notations* will be used: w is the word to be disambiguated (target word); $s_1, \ldots, s_K$ are possible senses for w; $c_1, \ldots, c_I$ are contexts of w in a corpus $\mathcal{C}$; $v_1, \ldots, v_J$ are words used as contextual features for the disambiguation of w.

Notice that the contextual features could be some attributes (morphological, syntactical, etc.), or they could be actual "neighboring" content words of the target word. The contextual features occur in a fixed position near w, in a window of fixed length, centered or not on w. 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 features are available, the entire sentence in which the target word occurs will represent the window of context.

The main assumptions of the *Naïve Bayes Model* are: (a) *the contexts* $\{c_i, i\}$ *in the corpus* $\mathcal{C}$ *are independent*, (b) *the contextual features are conditionally independent*. Hence, the likelihood of $\mathcal{C}$ is

$$P(\mathcal{C}) = \prod_{i=1}^{I} \sum_{k=1}^{K} P(s_k) \prod_{j=1}^{J} \left(P\left(v_j \mid s_k\right)\right)^{|v_j \ in \ c_i|}$$

The a posteriori probabilities of the senses, $P(s_k \mid c)$, are calculated by the Bayes formula.

The Bayes classifier (sometimes called the *Maximum A Posteriori classifier*) chooses the sense s' for which the a posteriori probability is maximal

$$s' = \arg\max_{k=1,\ldots,K} P(s_k \mid c)$$

We denote the parameters of the model with

$$\psi = (\alpha_1, \ldots, \alpha_K, \theta_{11}, \ldots, \theta_{KJ}),$$

where $P(s_k) = \alpha_k$, $k = 1, \ldots, K$, $\alpha_k \geq 0$ for all k, $\sum_{k=1}^{K} \alpha_k = 1$, and $P\left(v_j \mid s_k\right) = \theta_{kj}$, $k = 1, \ldots, K$, $j = 1, \ldots, J$, $\theta_{kj} \geq 0$ for all k and j, $\sum_{j=1}^{J} \theta_{kj} = 1$ for all $k = 1, \ldots, K$.

For estimating the parameters of the Naïve Bayes Model, the problem to be solved can be written as

$$\max \left(\sum_{i=1}^{I} \log \left(\sum_{k=1}^{K} \alpha_k \prod_{j=1}^{J} \left(\theta_{kj}\right)^{|v_j \ in \ c_i|} \right) \right) \tag{1}$$

with the constraints

$$\begin{aligned} &\sum_{k=1}^{K} \alpha_k = 1 \\ &\sum_{j=1}^{J} \theta_{kj} = 1 \ \text{ for all } \ k = 1, \ldots, K \end{aligned} \tag{2}$$

The optimization problem (1)&(2) can be solved only by iterative methods, the most used method being the Expectation-Maximization Algorithm (Dempster et al. 1977). Each iteration of the algorithm involves two steps:

- estimation of the missing data by the conditional expectation method (E-step)
- estimation of the parameters by maximization of the likelihood function for complete data (M-step)

The E-step calculates the conditional expectations given the current parameter values, and the M-step produces new, more precise parameter values. The two steps alternate until the parameter estimates in iteration $r + 1$ and r differ by less than a threshold ε.

The EM Algorithm is guaranteed to increase the likelihood $\log P(\mathcal{C})$ in each step. Therefore, two stopping criteria for the algorithm could be considered: (1) Stop when the likelihood $\log P(\mathcal{C})$ is no longer increasing significantly; (2) Stop when parameter estimates in two consecutive iterations no longer differ significantly.

Further on, we present the EM Algorithm for solving the optimization problem (1)&(2) :

The available data, called *incomplete data*, are given by the corpus $\mathcal{C}$. The *missing data* are the senses of the ambiguous words, hence they must be modelled by some random variables

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

The *complete data* consist of incomplete and missing data, and the corresponding likelihood of the corpus $\mathcal{C}$ becomes

$$P_{\text{complete}}(\mathcal{C}) = \prod_{i=1}^{I} \prod_{k=1}^{K} \left(\alpha_k \prod_{j=1}^{J} (\theta_{kj})^{|v_j \text{ in } c_i|} \right)^{h_{ik}}$$

The EM Algorithm starts with a *random initialization* of the parameters, denoted by

$$\psi^{(0)} = \left(\alpha_1^{(0)}, \ldots, \alpha_K^{(0)}, \theta_{11}^{(0)}, \ldots, \theta_{KJ}^{(0)} \right)$$

The *iteration* $(r+1)$ consists in the following two steps:

The E-step computes the missing data, based on the model parameters estimated at iteration r, as follows:

$$h_{ik}^{(r)} = \frac{\alpha_k^{(r)} \cdot \prod_{j=1}^{J} \left(\theta_{kj}^{(r)} \right)^{|v_j \text{ in } c_i|}}{\sum_{k=1}^{K} \alpha_k^{(r)} \cdot \prod_{j=1}^{J} \left(\theta_{kj}^{(r)} \right)^{|v_j \text{ in } c_i|}}, \quad i = 1, \ldots, I; k = 1, \ldots, K$$

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

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

$$\theta_{kj}^{(r+1)} = \frac{\sum_{i=1}^{I} |v_j \text{ in } c_i| \cdot h_{ik}^{(r)}}{\sum_{j=1}^{J} \sum_{i=1}^{I} |v_j \text{ in } c_i| \cdot h_{ik}^{(r)}}, \quad k = 1, \ldots, K; j = 1, \ldots, J$$

The stopping criterion for the algorithm is "Stop when parameter estimates in two consecutive iterations no longer differ significantly". It is well known that the EM iterations $\left(\psi^{(r)}\right)_r$ converge to the Maximum Likelihood Estimate $\widehat{\psi} = \left(\widehat{\alpha}_1, \ldots, \widehat{\alpha}_K, \widehat{\theta}_{11}, \ldots, \widehat{\theta}_{KJ}\right)$.

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 features v_j occurring in the context c. *Making the Naïve Bayes assumption and using the Bayes decision rule, we can decide s' if*

$$s' = \arg\max_{k=1,\ldots,K} \left(\log \widehat{\alpha}_k + \sum_{j=1}^{J} |v_j \text{ in } c| \cdot \log \widehat{\theta}_{kj} \right)$$

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 which fits well to models with missing data. It is equally based on previously existing discussions and reported results. The EM algorithm has been used with an underlying Naïve Bayes model in Pedersen and Bruce (1998) as well. Here it

is suggested that a combination of the EM algorithm and Gibbs Sampling might be appropriate. The EM algorithm has equally been used with a Naïve Bayes model in Gale et al. (1995), in order to distinguish city names from people's names. An accuracy percentage in the mid-nineties, with respect to *Dixon*, a name found to be quite ambiguous, was reported.

4 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 framework 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 more drastic in the case of unsupervised disambiguation,[2] 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 (Sect. 3) model, discussion among specialists focuses almost entirely on the issue of feature selection.

Two early approaches to word sense discrimination, Schütze's (1998) context group discrimination and Pedersen and Bruce's (1997, 1998) McQuitty's Similarity Analysis, rely on totally different sets of features and, in our view, still represent the main approaches to feature selection.

As commented in Pedersen (2006) 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 Pedersen and Bruce (1998) 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 Pedersen (2006) 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 (Bruce and Wiebe 1994; Ng and Lee 1996), Pedersen and Bruce (1998) make good use of them in unsupervised word sense disambiguation. They conducted an experimental evaluation relative to the 12-word sense-tagged corpus of Bruce and Wiebe (1994) as well as with the *line* corpus (Leacock et al. 1993). In the case of nouns, for which disambiguation results were the best, in combination with one of the considered

[2] See the experiments and test results with regard to unsupervised word sense disambiguation in Sect. 5.2 of the present paper.

[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.

feature sets, the obtained accuracy improved upon the most frequent sense by at least 10%. The most modest results (accuracy) were obtained when disambiguating the sense of the noun *line*.

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 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. 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) will be 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 will be 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 (Banerjee and Pedersen 2003), 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. A conclusion which is not surprising, as the examples contain words related syntagmatically to the target.

With respect to nouns, which represent the best developed portion of WordNet, previous studies (Banerjee and Pedersen 2003), performed for knowledge-based disambiguation, come to the conclusion that hyponym and meronym synsets are the most informative. However, we have equally taken into consideration hypernyms and holonyms. Tables 5 and 7 of Sect. 5.2 show the obtained disambiguation results when using various combinations of the mentioned types of WN synsets in the formation of the "disambiguation vocabulary".

Corresponding to adjectives, our disambiguation method has taken into account (Hristea and Popescu 2009) the *similarity* relation, which is typical of adjectives (and, in fact, only holds for adjective synsets contained in adjective clusters[4]). The *also-see* relation and the *attribute* relation have also been taken into account since these relations are considered most informative and have been found (Banerjee and Pedersen 2003) 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 (Banerjee and Pedersen 2002) 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

[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 (Fellbaum 1998). 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 (Fellbaum 1998).

of its antonymous sense in the window of context". Tables 8 and 9 of Sect. 5.2 show the obtained disambiguation results (Hristea and Popescu 2009) when using a "disambiguation vocabulary" in the formation of which all mentioned types of synsets have taken part. This is the vocabulary which has provided the best disambiguation results in the case of adjectives *common* and *public* (see Sect. 5.2.2). Disambiguation results were computed with and without antonym synsets participating in the disambiguation process.

In the case of verbs we suggest (Hristea 2009) additionally using, whenever possible, WN synsets indicated by the entailment relation,[5] and by the causal relation[6] which are typical of this part of speech. Table 10 of Sect. 5.2 shows the obtained disambiguation results (Hristea 2009) in the case of verb *help*.

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 the disambiguation process. In the case of our method each word (feature) contributes to the final score being assigned to a sense with a weight given by $P(v_j|s_k)$. Ths weight (probability) is not a priori established, but is learned by means of the EM algorithm.

5 Empirical evaluation

Tests concerning the proposed disambiguation method have so far concentrated on adjectives (Hristea and Popescu 2009). An experiment concerning verbs has also been performed (Hristea 2009). The present paper means to extend the discussed disambiguation method to the case of nouns, the part of speech for which the best disambiguation results have been mentioned by the literature so far. Conclusions regarding all these parts of speech will be presented.

In the case of nouns, which are placed under study for the first time in the present paper, the goal of the performed experiment is to compare results obtained by means of the new disambiguation method with those obtained by a classical unsupervised algorithm (one having an underlying Bayes model, which does not perform feature selection and which is trained with the EM algorithm). We shall equally compare our disambiguation results with those of Pedersen and Bruce (1998) 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 Sect. 4). It therefore performs feature selection, although in a completely different manner than that proposed by our method. With the necessity of performing feature selection of some type being obvious, disambiguation results concerning adjectives and verbs will be compared with those of Pedersen and Bruce (1998) only. In the case of all parts of speech test results will show that feature selection using a knowledge source of type WordNet is more effective in sense disambiguation than local type features are.

[5] The entailment relation between verbs resembles meronymy between nouns, but meronymy is better suited to nouns than to verbs (Fellbaum 1998).

[6] The causal relation (Fellbaum 1998) picks out two verb concepts, one causative (like *give*), the other what might be called the "resultative" (like *have*).

Table 1 Distribution of senses of *line*

Sense	Count
Product	2,218 (53.47%)
Written or spoken text	405 (9.76%)
Telephone connection	429 (10.34%)
Formation of people or things; queue	349 (8.41%)
An artificial division; boundary	376 (9.06%)
A thin, flexible object; cord	371 (8.94%)
Total count	4,148

5.1 Design of the experiments

5.1.1 Noun experiment

In the case of nouns we have used as test data the *line* corpus (Leacock et al. 1993). This corpus contains around 4,000 sense-tagged examples of the word *line* (noun) with subsets of their WordNet 1.5 senses. Examples are drawn from the WSJ corpus, the American Printing House for the Blind, and the San Jose Mercury. We have chosen the *line* data for our tests concerning nouns since this data set seems to have raised the greatest problems in the case of the Pedersen and Bruce (1998) approach to WSD, to which we shall be comparing the results of our own method. Pedersen and Bruce obtain the most modest disambiguation results in the case of the noun *line* (when testing for 5 different nouns).

The *line* data was created by Leacock et al. (1993) by tagging every occurrence of *line* in the selected corpus with one of 6 possible WordNet senses. These senses and their frequency distribution are shown in Table 1.

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

Two types of tests were performed in the case of the classical unsupervised algorithm. A first variant of testing involved a context window of size 5, which is a common size for WSD tests of this type. The second testing variant used a context window of size 25. This dimension was chosen in order for the two methods (the classical one and the newly proposed one) to be compared under the same conditions.

The newly introduced method generated a series of experiments that vary according to the specific sources used in establishing the so-called disambiguation vocabulary.

The overall source for creating the vocabulary was WordNet 3.0, which lists 30 different senses corresponding to the noun *line*. The *line* corpus is sense-tagged with subsets of WordNet 1.5 senses, namely with those senses listed in Table 1. Therefore a sense mapping of the initial (corpus) senses to those of the WN 3.0 database was necessary.

The sense "product" occurring in the *line* corpus has been mapped to the WN 3.0 synset having the $synset\_id$ 103671668 and containing the nouns {line, product line, line of products, line of merchandise, business line, line of business}. The sense "written or spoken text" occurring in the corpus corresponds to 3 WN 3.0 synsets, namely synset {note, short letter, line, billet} having the *id* 106626286, synset {line} having the *id* 107012534, and synset {line} having the *id* 107012979, respectively. The sense "telephone connection" occurring in the corpus corresponds to the WN 3.0 synset {telephone line, phone line, telephone circuit,

subscriber line, line} having the *id* 104402057. The sense "formation of people or things; queue" occurring in the corpus corresponds to 2 WN 3.0 synsets, namely synset {line} having the *id* 108430203 and synset {line} having the *id* 108430568, respectively. The sense "an artificial division; boundary" occurring in the corpus corresponds to the WN 3.0 synset {line, dividing line, demarcation, contrast} having the *id* 105748786. Finally, the sense "a thin, flexible object; cord" occurring in the corpus corresponds to the WN 3.0 synset {line} having the *id* 103670849.

Let us once again note that our disambiguation method is an unsupervised one and therefore does not require sense labels (but only the number of senses, as detailed in Sect. 5.2). Performing the presented sense mapping was necessary solely for establishing the restricted disambiguation vocabulary (relevant words).

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

The first performed experiment involving the disambiguation of the noun *line* establishes as relevant words forming the vocabulary all nouns of the 9 WN 3.0 synsets containing *line* which have been chosen as a result of sense mapping. Additionally, all content words occurring in the glosses of these synsets have been added to this vocabulary.[7] Within the following experiments information provided by the synsets related (through explicit relations existing in WN) to those containing the target word *line* has been successively added. Thus, the second performed experiment uses, along with all words occurring in the first one, the words existing in the hyponym and meronym synsets of the 9 synsets containing the target.[8] The third experiment uses all words occurring in the second one, to which all content words of all the hyponym and meronym synset glosses are added.[9] Within the next experiment the initially used vocabulary (first experiment) has been enriched by adding all words coming from all hyponym, hypernym, meronym and holonym synsets of the 9 synsets containing the target.[10] Finally, our last experiment uses all words involved in the previously described one, to which the content words occurring in the glosses of all synsets required by the previous experiment are added.[11]

5.1.2 Adjective experiment

In the case of adjectives we have used as test data the (Bruce et al. 1996) data 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 2, while Table 3 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

[7] Experiment referred to in Tables 5 and 7 as "Synonyms + Glosses".

[8] Experiment referred to in Tables 5 and 7 as "+Hyponyms + Meronyms".

[9] Experiment referred to in Tables 5 and 7 as "+Hyponyms + Glosses + Meronyms + Glosses".

[10] Experiment referred to in Tables 5 and 7 as "+Hyponyms + Hypernyms + Meronyms + Holonyms".

[11] Experiment referred to in Tables 5 and 7 as "+Hyponyms + Glosses + Hypernyms + Glosses + Meronyms + Glosses + Holonyms + Glosses".

Table 2 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	1,060

Table 3 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

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, 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 again 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 again necessary. According to this mapping, 4 WN 3.0 synsets took part in the disambiguation vocabulary corresponding to the adjective *common*, namely the synsets having the IDs 300492677[12], 302152473[13], 301673815[14] and 300970610[15], respectively. Both WN synsets corresponding to the adjective *public* and having the IDs 300493297[16] and 301861205[17], respectively were part of the same vocabulary when performing disambiguation tests relative to this adjective.

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

Each of the experiments involving the disambiguation of adjectives *common* and *public* have established (Hristea and Popescu 2009) as relevant words forming the vocabulary all

[12] This is synset {common} having the gloss 'belonging to or participated in by a community as a whole; public'.

[13] This is synset {common, mutual} having the gloss 'common to or shared by two or more parties'.

[14] This is synset {common} having the gloss 'to be expected; standard'.

[15] This is synset {common, usual} having the gloss 'commonly encountered'.

[16] This is synset {public} having the gloss 'affecting the people or community as a whole'.

[17] This is synset {public} having the gloss 'not private; open to or concerning the people as a whole'.

Table 4 Distribution of senses of *help*

Sense	Count
To enhance-inanimate object	78%
To assist-human object	22%
Total count	1,267

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[18] 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 interesting due to the "negative information" it can provide. The second performed experiment[19] eliminates from the disambiguation vocabulary all words brought in precisely by these antonym synsets.

5.1.3 Verb experiment

The newly proposed disambiguation method has been tested (Hristea 2009) in the case of verbs as well, since it is well known that the verb represents the part of speech which is the most difficult to disambiguate. Test results were equally compared to those obtained in Pedersen and Bruce (1998). Corresponding to verbs the (Bruce et al. 1996) data was used as test data once again. From this 12-word sense-tagged corpus the verb *help* was selected, out of a total of 4 sense-tagged verbs. This choice was again determined by the fact that *help* is a verb in the case of which disambiguation results are quite modest when using the Pedersen-and-Bruce-type local features. The distribution of senses corresponding to *help* that has been used in the performed experiments, as well as in Pedersen and Bruce (1998), is shown in Table 4.

The discussed disambiguation method can again generate a series of experiments that vary according to the specific sources used in establishing the disambiguation vocabulary. The overall source for creating this vocabulary was again WordNet 3.0, which lists 8 different senses corresponding to the verb *help*. In the case of this verb the disambiguation vocabulary was formed by taking into account all verbs of the 6 WN 3.0 synsets[20] containing *help* which have been chosen as a result of sense mapping, all content words occurring in the glosses and the associated example strings of these synsets, as well as all content words belonging to

[18] Referred to in Tables 8 and 9 as "all".

[19] Referred to in Tables 8 and 9 as "all-antonyms".

[20] These are the following:

- synset {help, aid} having the ID 200082081 and the gloss 'improve the condition of';
- synset {help} having the ID 200206998 and the gloss 'improve; change for the better';
- synset {serve, help} having the ID 201181295 and the gloss 'help to some food; help with food or drink';
- synset {avail, help} having the ID 201193569 and the gloss 'take or use';
- synset {help, assist, aid} having the ID 202547586 and the gloss 'give help or assistance; be of service';
- synset {help} having the ID 202555434 and the gloss 'contribute to the furtherance of'.

all WN-related synsets, their glosses and their corresponding example strings. This vocabulary can be regarded as an extended one, thus created in order to ensure greater coverage of the corpus instances for participation in the learning process, a requirement which has been proven more difficult to meet in the case of verbs.

5.2 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 (Pedersen and Bruce 1998) we have used the mapping that results in the highest classification accuracy.[21]

Test results are presented in Tables 5 and 7, 8, 9, 10. 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[22] and a threshold ε having the value 10^{-9}. Tables 5 and 7 (corresponding to nouns) also present, for enabling comparison, the results obtained by the classical algorithm, when using a context window of size 5 and 25, respectively. These results equally represent the average accuracy and standard deviation over 20 trials of the EM algorithm with a threshold ε having the value 10^{-9}.

Apart from accuracy, the following type of information is also included in Tables 5 and 7, 8, 9, 10: 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. Since the process of feature selection is based on the restriction of the disambiguation vocabulary, 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. Corresponding to these instances, the algorithm assigns the sense s_k for which the value of $P(s_k)$ (estimated by the EM algorithm) is maximum.

5.2.1 Test results concerning nouns

In the case of nouns, as can be seen in Table 5, the obtained disambiguation results when using an underlying Bayes model and applying the EM algorithm to the classical set of

[21] 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.

[22] The choice of this context window size is based on the suggestion of Lesk (1986) 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.

Table 5 Experimental results for 6 senses of *line*

Method	Number of features	Percentage of instances having only null features	Accuracy
Classic-5	4,700	0.0	.274 ± .02
Classic-25	9,932	0.0	.255 ± .02
Synonyms + glosses	73	45.7	.473 ± .02
+ hyponyms + meronyms	138	38.3	.478 ± .01
+ hyponyms + glosses	305	11.7	.454 ± .04
+ meronyms + glosses			
+ hyponyms + hypernyms	152	35.9	.465 ± .02
+ meronyms + holonyms			
+ hyponyms + glosses	358	8.5	.448 ± .05
+ hypernyms + glosses			
+ meronyms + glosses			
+ holonyms + glosses			

features, formed with the actual words occurring in the context window, are extremely modest. A possible cause of this failure is the great number of features used by the learning algorithm. This is also suggested by the fact that, when enlarging the context window from size 5 to size 25, the number of features increases from 4,700 to 9,932, which leads to a decrease in accuracy from 0.274 to 0.255. Let us note that accuracy in the same range (25–30%) is reported in Pedersen and Bruce (1998) when tests corresponding to all 6 senses of the *line* corpus are performed.

The first conclusion that results presented in Table 5 immediately lead to is that, whenever performing feature selection, accuracy increases substantially.

The best disambiguation result (0.478) was obtained in the case when the disambiguation vocabulary was formed with all WN synonyms occurring in all synsets that contain the target word, content words of the glosses corresponding to these synsets, as well as nouns coming from all their hyponym and meronym synsets. This is in accordance with previous findings of studies performed for knowledge-based disambiguation (Banerjee and Pedersen 2003) concluding that, in the case of nouns, hyponym and meronym synsets of those containing the target word are the most informative. The obvious conclusion is that making use of a knowledge base of type WordNet (in our case, for feature selection) substantially improves disambiguation results.

The same set of experiments, performed under the same conditions, has been conducted in the case of only 3 senses of *line*. The reason for performing this reduction from 6 to 3 senses was to verify to what extent the existence of a majority sense in the distribution of senses for *line*[23] influences the performances of our disambiguation method. The 3 chosen senses are listed in Table 6. We have thus obtained a more uniform distribution of the *line* senses. Additionally, the 3 senses we have selected coincide with the ones used in Pedersen and Bruce (1998) for the presented disambiguation experiment, which allows a straightforward comparison between the corresponding results.

[23] Sense "product" occurs in 53,47% of the *line* corpus instances.

Table 6 Distribution of the 3 chosen senses of *line*

Sense	Count
Telephone connection	429 (37.33%)
Formation of people or things; queue	349 (30.37%)
A thin, flexible object; cord	371 (32.28%)
Total count	1,149

Table 7 Experimental results for 3 senses of *line*

Method	Number of features	Percentage of instances having only null features	Accuracy
Classic-5	1,907	0.0	.280 ± .02
Classic-25	4,806	0.0	.248 ± .02
Synonyms + Glosses	30	49.7	.487 ± .03
+ hyponyms + meronyms	74	41.7	.513 ± .03
+ hyponyms + glosses	203	17.9	.570 ± .08
+ meronyms + glosses			
+ hyponyms + hypernyms	82	38.5	.498 ± .03
+ meronyms + holonyms			
+ hyponyms + glosses	229	15.1	.591 ± .06
+ hypernyms + glosses			
+ meronyms + glosses			
+ holonyms + glosses			

Test results for this case (3 senses of *line*) are presented in Table 7. As in the previous case (6 senses of *line*) the results obtained by the classical algorithm (without feature selection) are very modest, while performing feature selection consistently improves the results corresponding to each experiment. The maximum obtained accuracy (0.591) represents a consistent improvement over the maximum obtained in the previous case (0.478). This maximum accuracy was obtained corresponding to a disambiguation vocabulary formed with all words occurring in all synsets containing the target word, their hyponym, hypernym, meronym, and holonym synsets, to which all content words of all corresponding glosses are added. The explanation for obtaining the best result when using a larger number of explicit relations provided in WordNet could reside in the fact that, corresponding to the three chosen senses of *line*, no meronym synsets exist. This considerably reduces the disambiguation vocabulary that resulted in the best accuracy when all 6 senses of *line* were disambiguated.

We have compared our disambiguation method and corresponding results primarily to those of Pedersen and Bruce (1998) since both methods rely on an underlying Naïve Bayes model, use the EM algorithm for estimating model parameters[24] 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 and part of speech information near the target word, the present approach relies on WordNet and its rich set of semantic relations for performing

[24] Pedersen and Bruce (1998) also makes use of Gibbs sampling for parameter estimation, without results improving significantly.

Table 8 Experimental results for 3 senses of *common*

Method	Number of features	Percentage of instances having only null features	Accuracy
All	83	19.2	$.775 \pm .02$
All-antonyms	74	20.0	$.766 \pm .04$

Table 9 Experimental results for 3 senses of *public*

Method	Number of features	Percentage of instances having only null features	Accuracy
All	74	43.3	$.559 \pm .03$
All-antonyms	71	44.4	$.550 \pm .03$

feature selection. This places the disambiguation process at the border between unsupervised and knowledge-based techniques, but improves disambiguation accuracy consistently. Thus, the way in which our method performs feature selection brings the same disambiguation accuracy when testing for all 6 senses of *line* as that obtained in Pedersen and Bruce (1998) in the case of only 3 chosen senses of this target word (47%). The mentioned authors report that "accuracy degrades considerably, to approximately 25 to 30%, depending on the feature set" when testing for all 6 senses taken into consideration in the *line* corpus. When disambiguating only the same three chosen senses of *line*, the accuracy of our method is significantly higher (59%), being obtained with significant corpus coverage (the percentage of instances having only null features is 15.1). This clearly shows 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).

5.2.2 Test results concerning adjectives

Our method has been tested with respect to adjectives *common* and *public*, the latter being the one in the case of which Pedersen and Bruce (1998) obtain the most modest disambiguation results. Test results are presented in Table 8 (corresponding to adjective *common*) and Table 9 (corresponding to adjective *public*).

It can be noticed that the way in which our method performs feature selection brings a disambiguation accuracy of $.775 \pm .02$ in the case of the adjective *common*, while the highest accuracy obtained in Pedersen and Bruce (1998), corresponding to the same adjective and when estimating model parameters with the EM algorithm as well, is of $.543 \pm .09$. When leaving out antonym synsets the accuracy obtained by our method decreases to $.766 \pm .04$, which again represents a value significantly higher than the corresponding one of Pedersen and Bruce (1998). In the case of adjective *public* our method attains an accuracy of $.559 \pm .03$, which decreases to $.550 \pm .03$ when leaving out antonym synsets, with both values being higher than the corresponding one obtained in Pedersen and Bruce (1998): $.507 \pm .03$. 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).

Table 10 Experimental results for 2 senses of *help*

Method	Number of features	Percentage of instances having only null features	Accuracy
All	130	41.8	.671 ± .04

When analyzing the results presented in Tables 8 and 9 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.

5.2.3 Test results concerning verbs

Usage of the discussed disambiguation method has been exemplified and examined (Hristea 2009) in the case of verb *help*, corresponding to which Pedersen and Bruce obtain the most modest results (when estimating model parameters by means of the EM algorithm), out of 4 studied verbs. In the case of *help*, the best disambiguation accuracy attained in Pedersen and Bruce (1998) is .602 ± .03. Since, in this case, all WN synonyms corresponding to all chosen synsets will be verbs, which are unlikely to have multiple occurrences in the context window of the target word, and in order to ensure greater coverage of the corpus instances for participation in the learning process, an "extended disambiguation vocabulary" has been taken into account, as mentioned in Sect. 5.1.3. This vocabulary was created by using all verbs of the 6 WN 3.0 synsets containing *help* that have resulted after performing sense mapping, all content words occurring in the glosses and the associated example strings of these synsets, as well as all content words belonging to all WN - related synsets, their glosses and their corresponding example strings. The proposed disambiguation method was asked to perform discrimination among the two senses of *help* chosen in Pedersen and Bruce (1998) and presented in Table 4. As shown in Table 10, the obtained accuracy was .671 ± .04 with a number of 130 resulting features and with over 50% of the instances contributing to the learning process. This represents an improvement of the result obtained in Pedersen and Bruce (1998), although the verb is the most difficult to disambiguate part of spech.

Additionally, it is our belief that disambiguation results will improve corresponding to those verbs for which related synsets via the entailment and the causal relations, which are typical of verbs, exist. This was not the case of *help*, corresponding to which only hyponym and hypernym synsets were found. However, *help* has been chosen for the present discussion since it enables comparison with results found in Pedersen and Bruce (1998). In the case of verbs as well, feature selection using a knowledge source of type WordNet has once again proven to be more effective in disambiguation than local type features are.

6 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. The theoretical model was presented and its implementation was discussed. Special attention was paid to parameter estimation and feature selection, the two main issues of the model's implementation. The EM algorithm was recommended as suitable for parameter estimation. As far as feature selection is concerned, we have placed under survey a new method for performing it. The novelty of our method, proposed in Hristea and Popescu (2009), and extended here to the case of nouns, consists in using the semantic network WordNet as knowledge source for feature selection. The method makes ample use of the WordNet semantic relations which are typical of each part of speech. This places the disambiguation process at the border between unsupervised and knowledge-based techniques.

We have presented tests performed with our disambiguation method concerning nouns, adjectives, and verbs. The performed tests have shown that, in the case of all these parts of speech, feature selection using a knowledge source of type WordNet is more effective in disambiguation than local type features (like part-of-speech tags) are.

However true it may be that usage of WordNet for feature selection not only places our method at the border between unsupervised and knowledge-based techniques, but also reduces flexibility and language independence, let us keep in mind that knowledge-lean methods as the one proposed in Pedersen and Bruce (1998) can also require information that is not always available. Such knowledge-lean methods can equally have difficulties when asking for information like part of speech, for instance, especially if a part-of-speech tagger does not exist for the language under investigation.

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

Acknowledgments Work supported by the National University Research Council of Romania (the "Ideas" research programme, PN II-IDEI), Contract No. 659/2009. The authors express their deepest gratitude to Dr. Ted Pedersen for having provided the data set necessary for performing the presented tests and comparisons with respect to adjectives and verbs.

References

Agirre E, Edmonds P (eds) (2006) Word sense disambiguation. Algorithms and applications. Springer, The Netherlands

Banerjee S, Pedersen T (2002) An adapted Lesk algorithm for word sense disambiguation using WordNet. In: Proceedings of the third international conference on intelligent text processing and computational linguistics, Mexico City, February 17–23, pp 136–145

Banerjee S, Pedersen T (2003) Extended gloss overlaps as a measure of semantic relatedness. In: Proceedings of the eighteenth international joint conference on artificial intelligence, Acapulco, Mexico, pp 805–810

Bruce R, Wiebe J (1994). Word sense disambiguation using decomposable models. In: Proceedings of the 32nd meeting of the Association for Computational Linguistics, Las Cruces, New Mexico, pp 139–146

Bruce R, Wiebe J, Pedersen T (1996). The measure of a model. In: Proceedings of the conference on empirical methods in natural language processing, Philadelphia, PA, pp 101–112

Dempster A, Laird N, Rubin D (1977) Maximum likelihood from incomplete data via the EM algorithm. J Royal Stat Soc B 39(1):1–38

Fellbaum C (ed) (1998) WordNet: an Electronic Lexical Database. The MIT Press, Cambridge, MA

Gale WA, Church KW, Yarowsky D (1992) A method for disambiguating word senses in a large corpus. Comp Humanit 26(5–6):415–439

Gale WA, Church KW, Yarowsky D (1995) Discrimination decisions for 100,000—dimensional space. Ann Oper Res 55(2):323–344

Hristea F (2009) Recent advances concerning the usage of the Naïve Bayes Model in unsupervised word sense disambiguation. Int Rev Comput Softw 4(1):58–67

Hristea F, Popescu M (2009) Adjective sense disambiguation at the border between unsupervised and knowledge-based techniques. Fundam Inform 91(3–4):547–562

Leacock C, Towell G, Voorhees E (1993) Corpus-based statistical sense resolution. In: Proceedings of the ARPA workshop on human language technology, Princeton, New Jersey, pp 260–265
Lesk M (1986) Automatic sense disambiguation: how to tell a pine cone from an ice cream cone. In: Proceedings of the 1986 SIGDOC conference, New York, Association for Computing Machinery, pp 24–26
Miller GA (1990) Nouns in WordNet: a lexical inheritance system. Int J Lexicography 3(4):245–264
Miller GA, Beckwith R, Fellbaum C, Gross D, Miller K (1990) WordNet: an on-line lexical database. J Lexicography 3(4):234–244
Miller GA (1995) WordNet: a lexical database. Commun ACM 38(11):39–41
Miller GA, Hristea F (2006) WordNet nouns: classes and instances. Comput Linguist 32(1):1–3
Ng H, Lee H (1996) Integrating multiple knowledge sources to disambiguate word sense: an exemplar-based approach. In: Proceedings of the 34th annual meeting of the Society for Computational Linguistics, Santa Cruz, California, pp 40–47
Pedersen T, Bruce R (1997) Distinguishing word senses in untagged text. In: Proceedings of the second conference on empirical methods in natural language processing (EMNLP-2), Providence, Rhode Island, pp 197–207
Pedersen T, Bruce R (1998) Knowledge lean word-sense disambiguation. In: Proceedings of the 15th National conference on artificial intelligence, Madison, Wisconsin, pp 800–805
Pedersen T (2006) Unsupervised corpus-based methods for WSD. In: Agirre E, Edmonds P (eds) Word sense disambiguation Algorithms and Applications. Springer, The Netherlands pp 133–166
Schütze H (1998) Automatic word-sense discrimination. Comput Linguist 24(1):97–123