Artif Intell Rev (2014) 41:241–260
DOI 10.1007/s10462-011-9306-y

# Unsupervised word sense disambiguation with N-gram features

**Daniel Preotiuc-Pietro · Florentina Hristea**

Published online: 10 January 2012


**Abstract** The present paper concentrates on the issue of feature selection for unsupervised word sense disambiguation (WSD) performed with an underlying Naïve Bayes model. It introduces web N-gram features which, to our knowledge, are used for the first time in unsupervised WSD. While creating features from unlabeled data, we are "helping" a simple, basic knowledge-lean disambiguation algorithm to significantly increase its accuracy as a result of receiving easily obtainable knowledge. The performance of this method is compared to that of others that rely on completely different feature sets. Test results concerning nouns, adjectives and verbs show that web N-gram feature selection is a reliable alternative to previously existing approaches, provided that a "quality list" of features, adapted to the part of speech, is used.

**Keywords** Bayesian classification · The EM algorithm · Word sense disambiguation · Unsupervised disambiguation · Web-scale N-grams

## 1 Introduction

Word sense disambiguation (WSD) is a core research problem in computational linguistics and natural language processing (NLP), 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 in many cases essential, or even compulsory, either for natural language understanding, or for a wide range of applications such as: information retrieval, machine translation, speech processing, text processing etc.

D. Preotiuc-Pietro (✉)
Department of Computer Science, University of Sheffield,
Regent Court, 211 Portobello Street, Sheffield S1 4DP, UK
e-mail: daniel@dcs.shef.ac.uk

F. Hristea
Department of Computer Science, University of Bucharest, 14 Academiei Street, Sector 1,
010014 Bucharest, Romania
e-mail: fhristea@fmi.unibuc.ro

In the subfield of natural language processing (from the perspective of which we shall approach WSD within 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 by 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; Miller 1995; Fellbaum 1998).

In spite of the great number of existing disambiguation algorithms, the problem of WSD remains an open one, with three main classes of WSD methods being taken into consideration by the literature: supervised disambiguation, unsupervised disambiguation and knowledge-based disambiguation.

The present paper refers to unsupervised corpus-based methods for WSD. It concentrates on distributional approaches to unsupervised WSD that rely on monolingual corpora, with focus on the usage of the Naïve Bayes model as clustering technique. We are given $I$ sentences that each contain a particular polysemous word. Our goal 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; Hristea et al. 2008; Hristea 2009; Hristea and Popescu 2009), 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.

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 make a distinction 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 knowledge acquisition bottleneck (manual sense-tagging).

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. As already mentioned, 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, the main issue of the model's implementation. A novel method of performing knowledge-lean feature selection will be presented and discussed.

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 that 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. In fact, one can say that discussions concerning the implementation of the Naïve Bayes model for supervised / unsupervised WSD focus almost entirely on the issue of feature selection.

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

As commented in (Pedersen 2006, Schütze 1998) represents contexts in a high dimensional feature space that is created using a separate large corpus (referred to as the training corpus). While Schütze (1998) reduces dimensions by means of LSI/LSA, Pedersen and Bruce (1997) 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 (1998), Pedersen and Bruce (1998) 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.

More recently, Hristea et al. (2008) try to improve the disambiguation results previously obtained when performing unsupervised WSD based on an underlying Naïve Bayes model, by using the freely available semantic network WordNet (WN) as knowledge source for feature selection. Their method, intially tested for adjectives only, is extended to all main parts of speech and surveyed in (Hristea et al. 2008). The method makes ample use of the WordNet semantic relations that are typical of each part of speech, which places the disambiguation process at the border between unsupervised and knowledge-based techniques. The semantic network WordNet has been used as unique knowledge source for feature selection. Although not totally knowledge-lean, the full presentation of the method (Hristea et al. 2008) has once again reinforced the benefits of combining the unsupervised approach to the WSD problem with a knowledge source of type WordNet. Especially since we must 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.

The disambiguation results obtained in (Hristea et al. 2008) were compared to those of Pedersen and Bruce (1998) since both disambiguation methods use an algorithm of the same type i.e. unsupervised and based on an underlying Naïve Bayes model. However, the two compared algorithms perform feature selection in a completely different manner, as already specified. While Pedersen and Bruce (1998) use a restricted set of local features that include co-occurrence and part of speech information near the target word, Hristea et al. (2008) make use of WordNet for feature selection. The latter approach implements a Naïve Bayes model that uses as features *the actual words* occurring in the context window of the target and decreases the existing number of features by selecting a restricted number of such words, as indicated by WordNet. The size of the feature set is therefore reduced by performing knowledge-based feature selection. In the case of all parts of speech test results have shown (Hristea et al. 2008) that feature selection using a knowledge source of type WordNet is more effective in sense disambiguation than local type features are.

The present paper focuses on an entirely different way of performing feature selection, which is based on the intuition that the most frequently occurring words near the target could

give a better indication of the sense which is activated in context than words being semantically similar but which may not appear so often in the same context with the target word. The paper introduces web N-gram features which, to our knowledge, are used for the first time in unsupervised WSD.

The disambiguation method using N-gram features that we are presenting here is unsupervised and uses counts collected from the web in a simple way, in order to rank candidates. It does not require sense definitions or inventories, the only requirement is for these counts to be available. In this sense, as in the sense of Pedersen (2006), it is knowledge-lean.

Comparisons will be performed with the mentioned previous approaches that rely on an underlying Naïve Bayes model but on completely different feature sets, and especially with the approach of Hristea et al. (2008), which has so far proven to be the best, as far as feature selection for unsupervised WSD is concerned. Web N-gram feature selection will prove itself more efficient than local-type features usage, and test results concerning nouns, adjectives and verbs will recommend it as a reliable alternative to the other considered approach, provided that a "quality list" of features, adapted to the part of speech, is used.

This paper concerns feature selection for unsupervised WSD performed with a classical clustering method, hereby represented by the Naïve Bayes model, and introduces a new way of performing such feature selection. Specifically, we shall be providing a basic, simple knowledge-lean disambiguation algorithm (that relying on the Naïve Bayes model), with easily obtainable knowledge in the form of N-gram features. We thus create features from unlabeled data, a strategy which is part of a growing trend in natural language processing, together with exploiting the vast amount of data on the web (Keller and Lapata 2003). We ultimately compare totally different ways of feeding knowledge of various types to a knowledge-lean algorithm for unsupervised WSD. Our study will prove that a basic, simple knowledge-lean disambiguation algorithm can perform quite well when provided knowledge in an appropriate way, a remark also made by Ponzetto and Navigli (2010).

The rest of the article is organised as follows: Sect. 2 presents the underlying mathematical model, while Sect. 3 discusses feature selection and briefly presents the two entirely different approaches to this problem that our own approach will be compared to. Section 4 describes the novel type of feature selection proposed in this paper, Sect. 5 presents the experimental results and we conclude in Sect. 6.

## 2 The Naïve Bayes model

The algorithm for word sense disambiguation under study in this section 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 within a 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 but, instead, it combines 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.

### 2.1 The probability model of the corpus and the Bayes classifier

In order to formalize the described model, we shall present the probability structure of the corpus $C$. 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 probability structure of the corpus is based on one main assumption: *the contexts* $\{c_i, i\}$ *in the corpus* $\mathcal{C}$ *are independent*. Hence, the likelihood of $\mathcal{C}$ is given by the product

$$P(\mathcal{C}) = \prod_{i=1}^{I} P(c_i)$$

Let us note that this is a quite natural assumption, as the contexts are not connected, they occur at significant lags in $\mathcal{C}$.

On considering the possible senses of each context, one gets

$$P(\mathcal{C}) = \prod_{i=1}^{I} \sum_{k=1}^{K} P(s_k) \cdot P(c_i \mid s_k)$$

**A model with independent features** (usually known as the **Naïve Bayes Model**) assumes that the contextual features are conditionally independent. That is,

$$P(c_i \mid s_k) = \prod_{v_j \ in \ c_i} P\left(v_j \mid s_k\right) = \prod_{j=1}^{J} \left(P\left(v_j \mid s_k\right)\right)^{\left|v_j \ in \ c_i\right|},$$

where by $\left|v_j \ in \ c_i\right|$ we denote the number of occurrences of feature $v_j$ in context $c_i$. Then, the likelihood of the corpus $\mathcal{C}$ is

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

The parameters of the probability model with independent features are

$$\left\{P(s_k), k = 1, \ldots, K \ \text{ and } \ P\left(v_j \mid s_k\right), j = 1, \ldots, J, k = 1, \ldots, K\right\}$$

*Notation*:

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

With this notation, the likelihood of the corpus $\mathcal{C}$ can be written as

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

The well known Bayes classifier involves the a posteriori probabilities of the senses, calculated by the Bayes formula for a specified context $c$,

$$P(s_k \mid c) = \frac{P(s_k) \cdot P(c \mid s_k)}{\sum_{k=1}^{K} P(s_k) \cdot P(c \mid s_k)} = \frac{P(s_k) \cdot P(c \mid s_k)}{P(c)},$$

with the denominator independent of senses.

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

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

Taking into account the upper Bayes formula, one can define the Bayes classifier by the equivalent formula

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

Of course, when implementing a Bayes classifier, one has to estimate the parameters first.

### 2.2 Parameter estimation

Parameter estimation is performed by the Maximum Likelihood Method, for the available corpus $\mathcal{C}$. That is, one has to solve the optimization problem

$$\max\left(\log P(\mathcal{C}) \mid \left\{P(s_k), k = 1, \ldots, K \text{ and } P\left(v_j \mid s_k\right), j = 1, \ldots, J, k = 1, \ldots, K\right\}\right)$$

For the Naïve Bayes Model, the problem can be written as

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

with the constraints

$$\sum_{k=1}^{K} \alpha_k = 1$$

$$\sum_{j=1}^{J} \theta_{kj} = 1 \quad \text{for all } k = 1, \ldots, K$$

For unsupervised disambiguation, where no annotated training corpus is available, the maximum likelihood estimates of the parameters are constructed by means of the **Expectation-Maximization (EM) Algorithm.**

The optimization problem (1) can be solved only by iterative methods. The Expectation—Maximization Algorithm (Dempster et al. 1977) is a very successful iterative method, very well fitted for models with missing data.

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 $\varepsilon$.

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)

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

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

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

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

Hence, the log-likelihood for complete data is

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

Each M-step of the algorithm solves the maximization problem

$$\max \left( \sum_{i=1}^{I} \sum_{k=1}^{K} h_{ik} \left( \log \alpha_k + \sum_{j=1}^{J} |v_j\ in\ c_i| \cdot \log \theta_{kj} \right) \right) \tag{2}$$

with the constraints

$$\sum_{k=1}^{K} \alpha_k = 1$$

$$\sum_{j=1}^{J} \theta_{kj} = 1 \quad \text{for all } k = 1, \ldots, K$$

For simplicity, we denote the vector of parameters by

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

and notice that the number of independent components (parameters) is $(K-1) + (KJ-K) = KJ-1$.

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:

$$h_{ik}^{(r)} = P_{\psi^{(r)}}\left(h_{ik} = 1 \mid \mathcal{C}\right),$$

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

*The M-step* solves the maximization problem (2) and computes $\alpha_k^{(r+1)}$ and $\theta_{kj}^{(r+1)}$ as follows:

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

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

The stopping criterion for the algorithm is "Stop when parameter estimates in two consecutive iterations no longer differ significantly". That is, stop when

$$\left\| \psi^{(r+1)} - \psi^{(r)} \right\| < \varepsilon,$$

namely

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

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

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

Our choice of recommending usage of the EM algorithm for parameter estimation in unsupervised WSD with an underlying Naïve Bayes model is also based on previously existing discussions and reported results. The EM algorithm has equally been used for parameter estimation in (Pedersen and Bruce 1998; Hristea et al. 2008), to the results of which we shall be comparing our own disambiguation results.

## 3 Feature selection

When implementing the previously described mathematical model, discussion among specialists focuses almost entirely on the issue of feature selection. In what follows, we briefly present the two main existing types of feature selection that we shall be comparing our own approach to.

### 3.1 Pedersen and Bruce local type features

In performing unsupervised word sense disambiguation with an underlying Naïve Bayes model, Pedersen and Bruce (1997, 1998) define three different feature sets for each word and use them to formulate such a model describing the distribution of sense groups of that word. The feature sets taken into account were composed of various combinations of the following five types of features:

**Morphology** The feature denoted $M$ represents the morphology of the ambiguous word. In the case of nouns, for instance, $M$ is binary indicating singular or plural. For verbs, the value of $M$ indicates the tense of the verb and can have up to seven possible values. This feature is not used for adjectives.

**Part-of-speech** The features denoted $PL_i$ and $PR_i$ represent the part-of-speech (POS) of the word $i$ positions to the left or right, respectively, of the ambiguous word. Each POS feature can have one of five possible values: noun, verb, adjective, adverb or other.

**Co-occurrences** The features denoted $C_i$ are binary variables representing whether the $i^{th}$ most frequent content word in all sentences containing the ambiguous word occurs anywhere in the sentence being processed.

**Unrestricted collocations** The features denoted $UL_i$ and $UR_i$ are features with 20 possible values that indicate if one of the top 19 most frequent words occurs in position $i$ to the left ($UL_i$) or right ($UR_i$) of the target word.

**Content collocations** The features denoted $CL_1$ and $CR_1$ indicate the content word occurring in the position 1 place to the left or right, respectively, of the ambiguous word. In general, features ($CL_i, CR_i$) are identical to the unrestricted collocations, except they exclude function words and only represent content words.

All these features[1] are defined over a small contextual window (local-context) and are selected to produce low dimensional event spaces.

The three feature sets used in the experiments presented in (Pedersen and Bruce 1998) were designated ***A***, ***B*** and ***C*** and were formulated as follows:

***A***: $M, PL_2, PL_1, PR_1, PR_2, C_1, C_2, C_3$
***B***: $M, UL_2, UL_1, UR_1, UR_2$
***C***: $M, PL_2, PL_1, PR_1, PR_2, CL_1, CR_1$

It is our belief that the most interesting aspect of the described approach is represented by the choice of such types of features and feature sets in order to formulate a Naïve Bayes model. However, as the authors note in (Pedersen and Bruce 1998) "while frequency-based features, such as those used in this work, reduce sparsity, they are less likely to be useful in distinguishing among minority senses".

Pedersen and Bruce consider nouns, verbs and adjectives as possible target words in the discrimination task, and explore the use of several different combinations of features.

[1] For more details concerning these types of features, feature sets, and their usage, see (Pedersen and Bruce 1998).

The two mentioned authors conducted an experimental evaluation in (Pedersen and Bruce 1998) relative to the 12—word sense—tagged corpus of Bruce et al. (1996) as well as with the *line* corpus (Leacock et al. 1993).

The obtained performance when using the described type of local-context features is relatively low. The best results were obtained in the case of nouns, where in combination with a specific feature set the obtained accuracy improved upon the most frequent sense by at least 10%. The most modest results (accuracy) were obtained in the case of the noun *line*.

No feature set resulted in greater accuracy than the most frequent sense for verbs and adjectives, a result that we find quite discouraging. In the case of nouns McQuitty's method performed better. In combination with feature set **B** it improved upon the most frequent sense by at least 10%. Pedersen and Bruce found that feature set **B** performs best for nouns, while feature set **C** performs best for both adjectives and verbs. Their disambiguation results were compared to those reported in (Hristea et al. 2008; Hristea 2009) where knowledge-based feature selection was performed. In the case of all parts of speech, test results have shown that feature selection using a knowledge source of type WordNet is more effective in sense disambiguation than local type features are.

### 3.2 WordNet-based feature selection

The approach to WSD of Hristea et al. (2008) relies on a set of features formed by the actual words occurring near the target word (within the context window) and reduces the size of this feature set by performing knowledge-based feature selection that relies entirely on WordNet. The WN semantic network provides 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, the words occurring in synsets related (through explicit relations provided in WordNet) to those containing the target word have also been considered as part of the vocabulary used for disambiguation. Synsets and relations were restricted to those associated with the part of speech of the target word. The content words of the glosses of all types of synsets participating in the disambiguation process, using the corresponding example strings as well, have been equally taken into consideration. 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, have come to the conclusion that *hyponym* and *meronym synsets* are the most informative. However, the mentioned authors have equally taken into consideration (Hristea et al. 2008) *hypernyms* and *holonyms*. Various combinations of the mentioned types of WN synsets have been used (Hristea et al. 2008) in the formation of the so-called "disambiguation vocabulary". The best disambiguation result has been obtained (Hristea et al. 2008) in the following case: the disambiguation vocabulary was formed with all WordNet synonyms occurring in all synsets that contain the target word, content words of the glosses and example strings corresponding to these synsets, as well as nouns coming from all their hyponym and meronym synsets (tests conducted for the noun *line*).

Corresponding to adjectives, the same disambiguation method has taken into account (Hristea and Popescu 2009; Hristea et al. 2008) the *similarity* relation, which is typical of this part of speech (and which only holds for adjective synsets contained in adjective clusters). 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 of its antonymous sense in the window of context". The disambiguation vocabulary providing the best test results (Hristea et al. 2008) was one in the formation of which all mentioned types of synsets have taken part (together with their glosses and associated example strings). Disambiguation results were computed (Hristea and Popescu 2009; Hristea et al. 2008) with and without antonym synsets participating in the disambiguation process. Results were always in favor of antonym participation (tests conducted for adjectives *common* and *public*).

In the case of verbs it was suggested (Hristea et al. 2008; Hristea 2009) to additionally use, whenever possible, WN synsets indicated by the *entailment* relation and by the *causal* relation which are typical of this part of speech. As in the case of adjectives, corresponding to verbs, the disambiguation vocabulary can be regarded as an extended one. It is formed by taking into account all verbs of the synsets containing the target word, 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 (tests conducted for the verb *help*).

As a result of using only those words indicated as being relevant by WordNet, a much smaller vocabulary was obtained, and therefore a much smaller number of features were taking part in the disambiguation process. In the case of this method (Hristea et al. 2008; Hristea 2009; Hristea and Popescu 2009), each word (feature) contributes to the final score being assigned to a sense with a weight given by $P(v_j \mid s_k)$. This weight (probability) is not a priori established, but is learned by means of the EM algorithm.

## 4 The web as a corpus

With respect to feature selection we wish to have those words that are the most relevant and distinctive for the target word. So, it is intuitive to think that these words are the ones that co-occur most often with the target word. These words can be found by searching and performing an estimate over large corpora and the largest corpora available is the whole Web itself.

While the web provides an imense linguistic resource, collecting and processing data at web-scale is very timeconsuming. Previous research has relied on search engines to collect online information, but an alternative to this that has been developed is to use the data provided in an N-gram corpus. An N-gram corpus is an efficient compression of large amounts of text as it states how often each sequence of words (up to length N) occurs.

The feature selection method that we introduce in this paper makes use of the Google Web 1T 5-gram Corpus Version 1.1, introduced in (Brants and Franz 2006), that contains English word N-grams (with $N = 5$) and their observed frequency counts, calculated over 1 trillion

words from the web collected by Google in January 2006. The text was tokenized following the Penn Treebank tokenization, except that hyphenated words, dates, email addresses and URLs are kept as single tokens. The sentence boundaries are marked with two special tokens <S> and </S>. Words that occurred fewer than 200 times were replaced with the special token <UNK>. The data set has a N-gram frequency cutoff, that is N-grams that have a count that is less than 40 are discarded.

This corpus has been used in a variety of NLP tasks with good results. (Bergsma et al. 2009) presents a unified view of using web-scale N-gram models for lexical disambiguation and uses the counts of 2–5 grams in a supervised method on the task of preposition selection, spelling correction or non-referential pronoun detection. In (Bergsma et al. 2010) web-scale N-gram data was used for supervised classification on a variety of NLP tasks such as: verb part-of-speech disambiguation, prenomial adjective ordering or noun compound braketing. Islam and Inkpen (2009) used the N-gram data for spelling correction, while Chang and Clark (2010) made use of the data to check the acceptability of paraphrases in context.

Web-scale N-gram counts have never been used, so far, for unsupervised word sense disambiguation or as a mean of feature selection.

In order to find the most frequent words that co-occur with the target word within a distance of N-1 words, we must take into consideration the N-grams in which the target word occurs. Thus, we can build different feature sets depending on the size of N and on the number of words to include in the feature set. These sets will be referred using the following convention: **n-w-t** represents the set containing the top **t** words occuring in **n**-grams together with the word **w**.

For example, *5-line-100* is the set constituted by the most frequent 100 (stemmed) words that co-occur in the Web with the word *line* within a distance of, at most, 4 words.

In order to build the feature set corresponding to the top $t$ words occurring in N-grams of size $n$ with the target word $w$, (n-w-t), we have used the following processing directions:

- We have lowercased every occurrence in the N-gram corpus and have combined the counts for identical matches;
- For every number $k$ ($k < n$), we have built a list of words and counts, each representing word counts occurring at a distance of exactly $k$ on each side of the target word;
- We have merged the counts from all *n-1* lists to get a complete list of words and counts that co-occur in a context window of size *n-1* with the target word $w$;
- We have removed the numbers, the punctuation marks, the special tokens (eg. <s>, <unk>), the words starting with special characters or symbols and the stopwords from the list;
- We have performed stemming using the Porter Stemmer on each feature set, merging counts for similar words whenever the case;
- we have sorted the word and counts pairs in descending order of their counts and have extracted the top $t$ words.

We mention that we couldn't use the lists for lower order N-grams in building higher order N-gram lists because of the cutoff value for N-gram counts, set at 40 for the Google corpus. Despite this small design problem of the N-gram corpus, the lists derived in both ways would be very similar.

We should also keep in mind the fact that, while in the context window only content words exist, within the N-grams stopwords may also occur. So, it is not guaranteed that the N-grams show the counts of words appearing in a context window of N-1. We have chosen to eliminate stopwords because they appear way too often in the corpora and, by using them as features,

the model tends to put too much weight on these, as opposed to the content words that are really indicative of the word sense.

Despite the fact that the target words and the dataset we refer to in the experiments are in English, the feature selection method we propose is language independent and can be applied with no extra costs to other languages for which we know or can estimate N-gram counts from large data. Recently, Google has released Web 1T 5-gram, 10 European Languages Version 1 (Brants and Franz 2009) consisting of word N-grams and their observed frequency counts for other ten European languages: Czech, Dutch, French, German, Italian, Polish, Portuguese, Romanian, Spanish and Swedish. The N-grams were extracted from publicly accessible web pages from October 2008 to December 2008 using the same conventions as for the English data set, with only the data being approximately 10 times smaller. Thus, our method can be used with no changes whatsoever to extract features for performing sense disambiguation corresponding to these languages as well.

Using a web-scale N-gram corpus implies performing counts that take into account all the possible senses of the target word. Automatically, when computing these counts, high frequency senses will have more words indicative of those senses than low frequency senses have. If our disambiguation setting is restricted to a specific domain (eg. medicine), our method of feature extraction could be used with a N-gram corpus derived from large corpora of texts in that domain.

## 5 Experimental results

### 5.1 Corpora

The present paper aims to test the newly proposed feature sets for the three main parts of speech: nouns, adjectives and verbs. We shall be drawing conclusions regarding each of these parts of speech.

In order to compare our results with those of other previous studies (Pedersen and Bruce 1998; Hristea et al. 2008) that have presented the same model, trained with the EM algorithm but using different methods of feature selection, we hereby try to disambiguate the same target words using the same corpora.

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 one of the 6 possible WordNet 1.5 senses. Examples are drawn from the WSJ corpus, the American Printing House for the Blind, and the San Jose Mercury. The description of the senses and their frequency distribution are shown in Table 1.

**Table 1** Distribution of senses of *line*

| Sense | Count | Percentage |
|---|---|---|
| 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 | 100 |

**Table 2** Distribution of senses of *common*

| Sense | Count | Percentage |
|---|---|---|
| As in the phrase "common stock" | 892 | 84 |
| Belonging to or shared by 2 or more | 88 | 8 |
| Happening often; usual | 80 | 8 |
| Total count | 1.060 | 100 |

**Table 3** Distribution of senses of *public*

| Sense | Count | Percentage |
|---|---|---|
| Concerning people in general | 440 | 68 |
| Concerning the government and people | 129 | 19 |
| Not secret or private | 90 | 13 |
| Total count | 659 | 100 |

**Table 4** Distribution of senses of *help*

| Sense | Count | Percentage |
|---|---|---|
| To enhance-inanimate object | 990 | 78 |
| To assist-human object | 279 | 22 |
| Total count | 1.269 | 100 |

In (Pedersen and Bruce 1998; Hristea et al. 2008), tests are also performed for only 3 senses of line. We will not perform this comparison as our method is not relying on sense inventories so we cannot distinguish and take out the words that co-occur with the specific senses represented in the test set.

In the case of adjectives and verbs we have used as test data the corpus introduced in (Bruce et al. 1996) that contains twelve words taken from the ACL/DCI Wall Street Journal corpus and tagged with senses from the Longman Dictionary of Contemporary English. Tests will be conducted for two adjectives, *common* and *public*, the latter being the one corresponding to which Pedersen and Bruce (1998) obtain the worst disambiguation results. The senses of *common* and *public* that have been taken into consideration and their frequency distribution are shown in Tables 2 and 3 respectively. In order to compare our results to those of Pedersen and Bruce (1998), Hristea et al. (2008) we have also taken into account only the 3 most frequent senses of each adjective, as was the case in those studies.

For verbs, the part of speech which is known as being the most difficult to disambiguate, we have performed tests corresponding to the verb *help* while considering the most frequent two senses of this word. The definition of the senses and the frequency distribution are presented in Table 4.

In order for our experiments to be conducted, the data set was preprocessed as follows: the stopwords, words with special characters and numbers were eliminated and stemming was applied to all remaining words, using the same Porter Stemmer as in the case of stemming the lists of feature words.

### 5.2 Tests

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 method. These sense groups must be mapped to sense tags in order to evaluate system performance. As in the previously mentioned studies, is order to enable comparison, we have used the mapping that results in the highest classification accuracy.

In the case that we can't find any of the words belonging to the feature set in the context window of the target, as in (Hristea et al. 2008), our method assigns the instance to the cluster that has the greatest number of assignments. If the target word has a dominant sense, which is the case with all our test target words, lower coverage will determine an increase in the performance of the method when results are below the most frequent sense baseline (a very high one in the case of unsupervised WSD using the same underlying mathematical model). With respect to this, we also define coverage as the percentage of instances in which at least one feature word occurs in the context window and, so, the assigment is performed by our Naïve Bayes classifier as opposed to a most frequent sense one.

We show results that couple accuracy with coverage. We use a context window with varying size around the target word, the coverage for a feature set increasing accordingly with the enlargement of the window size.

As in (Hristea et al. 2008) each result represents the average accuracy obtained by the disambiguation method over 20 random trials while using a fixed threshold $\epsilon$ having the value $10^{-9}$.

We show the most significant test results that were obtained in case of all the 3 different parts of speech.

Within the graphs, our results are designated by solid lines with different markers indicating the various parameters ($n$ or $t$) that we have used. The context window sizes vary and are listed in the corresponding text for each part of speech. The Hristea et al. (2008) method is presented with a dashed line and always uses a context window of size 25. The variation in coverage is due to the different type of WordNet relations that have been used, resulting in a different number of feature words. The results of the Pedersen and Bruce (1998) method are presented as well. We notice that here we always have just one value, corresponding to a 100% coverage and to a size of 5 or 25 of the context window. This is due to the fact that the method of feature selection takes into consideration all the words in the vocabulary. Therefore, in this case, there are no contexts with null features. In each graph, corresponding to each of the other two previous methods, and in order to allow an easier visual comparison, we have drawn a dotted black line to illustrate the highest accuracy obtained for that word with the respective method.

### *5.2.1 Nouns*

In the case of the word *line* results are presented in Fig. 1.

The best results were obtained by using the most frequent words appearing in 5-grams with *line*, although results with a lower $n$ were only slightly worse.

Test results are presented for context windows of size 4, 5, 10, 15 and 25 corresponding to each feature set. We observe the largest difference in favour of our feature selection method as resulting in an accuracy of 54.7% (for context window 5 and feature set 5-line-100) as compared to 47.8% for a similar coverage in (Hristea et al. 2008). For the feature sets *5-line-100* and *5-line-200*, our tests show better performances than any of the results of Hristea et al. (2008) and better, by a wide margin, than those of Pedersen and Bruce (1998). For some experiments, the method outperforms the most frequent sense baseline which, in this case, is situated at 53.47%.

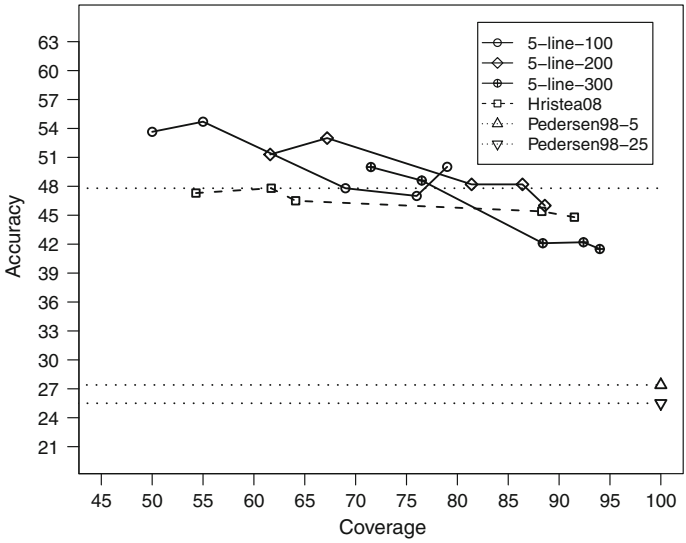


**Fig. 1** Results for feature sets 5-line

The graph also shows that by increasing too much the number of features (5-line-300), the performance of the system decreases. This performance decreases even more when considering even larger feature sets ($t = 500$ or 1000—not shown on the graph for clarity).

We observe that when feature selection is performed in the case of noun disambiguation, increasing the size of the context window (thus bringing more features into the process) does not bring improvements to the disambiguation results (taking into consideration the coverage-accuracy trade-off), as stated in other studies. Another interesting aspect is that, by every step in extending the context window, the coverage increases significantly. This remark is not valid, as we shall see, in the case of adjectives and verbs.

The obtained results confirm the intuition that, in order to disambiguate a noun, the information in a wide context is useful and can contribute to the disambiguation process. Features taken from wider contexts are also good indicators for disambiguation.

### *5.2.2 Adjectives*

With respect to adjectives, we have considered the disambiguation of the polysemous words *common* and *public*. Test results are shown in Figs. 2 and 3 respectively.

The best results were achieved by using the most frequent words appearing in bigrams with *common* and in 3-grams with *public* (although results with bigrams for *public* were close in terms of accuracy).

In the case of adjective *common* the results are presented for context windows of size 1, 2, 3, 4, 5 and 10. We observe the largest difference in favour of our feature selection method as resulting in an accuracy of 87.0%, compared to 77.5%, the best result obtained in (Hristea et al. 2008). Again, almost all scores (16 out of 18 shown) are higher than the ones of the Hristea et al. (2008) method, with almost half of them exceeding the most frequent sense baseline (at 84.0% in this case).

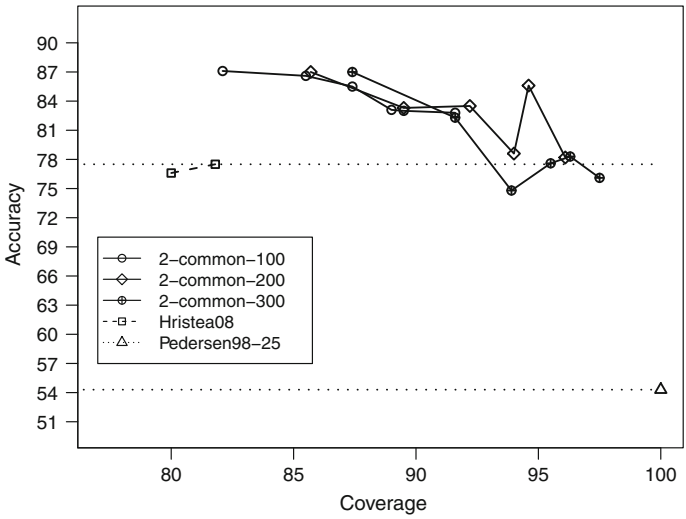


**Fig. 2** Results for feature sets 2-common

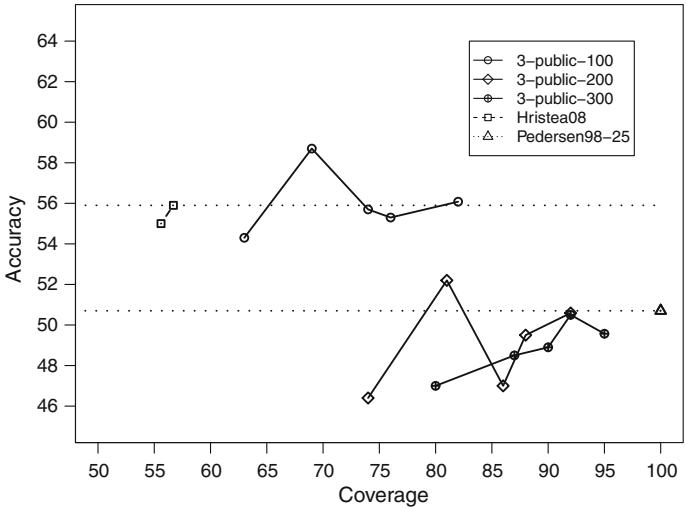


**Fig. 3** Results for feature sets 3-public

Corresponding to the adjective *public* test results are presented for context windows of size 2, 3, 4, 5 and 10. Our best result is 58.7% accuracy compared to 55.9% obtained with a much smaller coverage in (Hristea et al. 2008).

We must keep in mind that, as we move to the right of the graph (increasing coverage), the results are more significant, because the bias of choosing the most frequent sense baseline for contexts with no features is reduced, due to the fact that the baseline has a very high value (84 and 68% respectively).

For both adjectives, we observe that just by taking the most frequent 100 words in bigrams or trigrams and a very narrow context window (starting with size 1) we already obtain a very high coverage, that increases at a low rate together with the enlargement of the context window. This corresponds to the linguistic argument that an adjective will appear

together with the word it modifies, the latter representing the most frequent and important attribute when disambiguating the respective adjective. Results with wider N-grams were inferior by a distinctive margin.

### *5.2.3 Verbs*

Corresponding to the verb *help* test results are shown in Fig. 4.

Interestingly enough, the best results were achieved by using the top 100 words regardless of the order of the N-grams. Our top result was 73.1% when using words from 4-grams and a context window of size 15, as compared to a maximum of 67.1% in (Hristea et al. 2008), obtained with a similar coverage. Out of our 12 results, 11 were better than those in (Hristea et al. 2008), confirming the reliability of disambiguating using these feature sets.

The results are presented for context windows of size 10, 15 and 25 respectively, as coverage is too low for smaller context windows. We can conclude that, in general, coverage for verbs is very low compared to the case of nouns and adjectives and that it increases by a very low margin with the enlargement of the context window.

This is also very linguistically intuitive because verbs usually appear in very different contexts. This makes feature selection more difficult and is the main reason why most studies conclude that this is the hardest to disambiguate part-of-speech.

As we are shown from the results, corresponding to all parts of speech, we can restate the fact that, by taking more, less related words (increasing $t$), the accuracy drops, a fact which emphasizes the need for a "quality list of features". The presented feature selection method obtains very high results compared to Pedersen and Bruce (1998) in all tests, good results compared to Hristea et al. (2008) and sometimes exceeds the most frequent sense baseline, which is a high baseline to achieve using this methematical model.

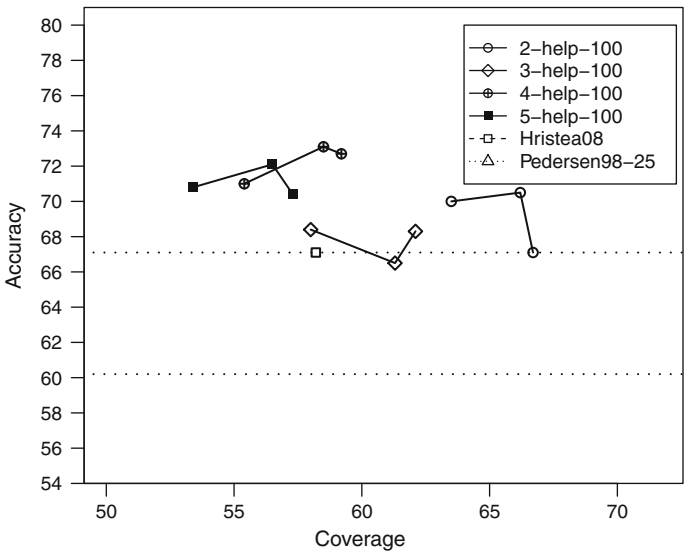


**Fig. 4** Results for feature sets help-100

## 6 Conclusions

This paper has focused on the issue of feature selection for unsupervised WSD performed with an underlying Naïve Bayes model. It has introduced a novel method of performing such feature selection that relies on using web scale N-gram counts.

This newly proposed feature selection method is based on the intuition that the most frequently occurring words near the target can give us a better indication of the sense which is activated than words being semantically similar that may not appear so often in the same context with the target word.

The disambiguation method using N-gram features that we have presented here is unsupervised and uses counts collected from the web in a simple way, in order to rank candidates. It creates features from unlabeled data, a strategy which is part of a growing trend in natural language processing, together with exploiting the vast amount of data on the web. Thus, the method does not rely on sense definitions or inventories. It is knowledge-lean in the sense that it just requires the existence or the possibility to estimate N-gram counts for the target language corresponding to which the disambiguation process takes place. No information regarding the actual word senses is used at any stage of the process.

Comparisons have been performed with previous approaches that rely on completely different feature sets. In the case of all studied parts of speech, test results were better, by a wide margin, than those obtained when using local-type features. They have also indicated a reliable alternative to WordNet feature selection, which has provided the best results so far, as reported in other studies (Hristea et al. 2008; Hristea and Popescu 2009; Hristea 2009) that have used the same underlying Naïve Bayes model.

The experiments conducted for all three major parts of speech (nouns, adjectives, verbs) have provided very different results, depending on the feature sets that were used. These results are in agreement with the linguistic intuitions and indicate the necessity of taking into consideration feature sets that are adapted to the part of speech which is to be disambiguated.

Last but not least, the presented method has once again proven that a basic, simple knowledge-lean disambiguation algorithm can perform quite well when provided knowledge in an appropriate way.

**Acknowledgments** The work of author Florentina Hristea was supported by the National University Research Council of Romania (the "Ideas" research program, PN II IDEI), Contract No. 659/2009

## References

Agirre E, Edmonds PG (2006) Word sense disambiguation: algorithms and applications (Text, Speech and Language Technology). Springer, Dordrecht

Banerjee S, Pedersen T (2002) An adapted lesk algorithm for word sense disambiguation using wordnet. In: Proceedings of the third international conference on computational linguistics and intelligent text processing, CICLing '02, 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, pp 805–810

Bergsma S, Lin D, Goebel R (2009) Web-scale N-gram models for lexical disambiguation. In: Proceedings of the 21st international jont conference on artifical intelligence, pp 1507–1512

Bergsma S, Pitler E, Lin D (2010) Creating robust supervised classifiers via web-scale N-gram data. In: Proceedings of the 48th annual meeting of the association for computational linguistics, ACL '10, pp 865–874

Brants T, Franz A (2006) Web 1T 5-gram corpus version 1.1. Technical report, Google research

Brants T, Franz A (2009) Web 1T 5-gram, 10 european languages version 1. Technical report, Linguistic data consortium, Philadelphia

Bruce R, Wiebe J, Pedersen T (1996) The measure of a model. CoRR, cmp-lg/9604018
Chang C-Y, Clark S (2010) Linguistic steganography using automatically generated paraphrases. In: Human language technologies: the 2010 annual conference of the North American chapter of the association for computational linguistics, HLT '10, pp 591–599
Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc B 39:1–38
Fellbaum Ce (1998) Wordnet: an electronic lexical database. The MIT Press, Cambridge
Gale W, Church K, Yarowsky D (1992) A method for disambiguating word senses in a large corpus. Comput Humanit 26:415–439
Hristea F (2009) Recent advances concerning the usage of the naive bayes model in unsupervised word sense disambiguation. Int Rev Comput Softw 4:58–67
Hristea F, Popescu M (2009) Adjective sense disambiguation at the border between unsupervised and knowledge-based techniques. Fundam Inf 91:547–562
Hristea F, Popescu M, Dumitrescu M (2008) Performing word sense disambiguation at the border between unsupervised and knowledge-based techniques. Artif Intell Rev 30:67–86
Islam A, Inkpen D (2009) Real-word spelling correction using google web it 3-grams. In: Proceedings of the 2009 conference on empirical methods in natural language processing, vol 3, EMNLP '09, pp 1241–1249
Keller F, Lapata M (2003) Using the web to obtain frequencies for unseen bigrams. Comput Linguist 29:459–484
Leacock C, Towell G, Voorhees E (1993) Corpus-based statistical sense resolution. In: Proceedings of the workshop on human language technology, HLT '93, pp 260–265
Miller GA (1990) Nouns in wordnet: a lexical inheritance system. Int J Lexicogr 3:245–264
Miller GA (1995) Wordnet: a lexical database for English. Commun ACM 38:39–41
Miller GA, Beckwith R, Fellbaum C, Gross D, Miller K (1990) Wordnet: an on-line lexical database. Int J Lexicogr 3:235–244
Pedersen T (2006) Unsupervised corpus-based methods for wsd. In: Word sense disambiguation: algorithms and applications. In: Agirre E, Edmonds P (eds) Springer, Dordrecht, pp 133–166
Pedersen T, Bruce R (1997) Distinguishing word senses in untagged text. In: Proceedings of the second conference on empirical methods in natural language processing, pp 197–207
Pedersen T, Bruce R (1998) Knowledge lean word-sense disambiguation. In: Proceedings of the fifteenth national conference on artificial intelligence, AAAI Press, pp 800–805
Ponzetto SP, Navigli R (2010) Knowledge-rich word sense disambiguation rivaling supervised systems. In: Proceedings of the 48th annual meeting of the association for computational linguistics, ACL '10, pp 1522–1531
Schütze H (1998) Automatic word sense discrimination. Comput Linguist 24:97–123