Comparative study of statistical word sense discrimination techniques

Marie-Catherine de Marneffe, Pierre Dupont
mcdm@info.ucl.ac.be, pdupont@info.ucl.ac.be

Computing Science Department, INGI – UCL
B-1348 Louvain-la-Neuve - Belgium

Abstract

Word sense discrimination aims at automatically determining which instances of an ambiguous word share the same sense. A fully unsupervised technique based on a vector representation of word senses was proposed by Schütze (Schütze, 1998). While the original model was assumed to be Gaussian, practical results were only reported for an approximated model making hard decisions between sense clusters. We show in the present study that a real Gaussian model provides a significant accuracy improvement while remaining fully tractable. An alternative discrete naïve Bayes model was presented in (Manning and Schütze, 1999). We propose here a description of both models in a unified statistical formalism in order to stress the similarities and differences between both approaches. Several practical experiments are conducted on the New York Times News 1997 corpus. They illustrate the respective advantages of various approaches trading off discrimination accuracy and computation time. We also show the interest of a global selection of content words to characterize the context of an ambiguous instance in the naïve Bayes model.

Keywords: Word sense disambiguation, Discrimination techniques, Naïve Bayes, K-means, Expectation-Maximization algorithm.

1 Introduction

The purpose of automatic word sense disambiguation is to determine the exact sense of an instance of an ambiguous word according to its particular use. Disambiguation can be useful in principle in any linguistic application where word sense matters such as automatic translation, text categorization, speech understanding, *etc.*

1.1 Word sense disambiguation techniques

Word sense disambiguation techniques can be divided into three broad categories: supervised techniques, dictionary-based (or thesaurus-based) and unsupervised techniques. All these techniques use the possible *senses* of the ambiguous word, the *contexts* of the instances of the ambiguous word and some sense *informants.*

Supervised techniques require a semantically tagged corpus, which serves as training corpus, in which each ambiguous instance w is correctly labeled with a semantic tag. The possible *senses* are defined by the set of semantic tags present in the corpus. The *contexts* consist of a window around instances of w, possibly limited to the syntactic group of w, and *informants* are the words belonging to those context windows. For example, Gale *et al.* use a *naïve Bayes classifier* to disambiguate words: the training corpus enables to assign to each informant the probability that it induces a sense (Gale *et al.* , 1992). Brown *et al.* propose an *information theoretic approach* which gives a sense to an ambiguous word as used for translation (Brown *et al.* , 1991). This technique determines the different values of the best informant. For instance, "prendre la voiture" in French is translated in English by "to *take* the car" and "prendre une décision" by "to *make* a decision". Here the informant is the verb object. Once the informant and its values have been found, an algorithm based on *mutual information* is applied to determine which informant value induces a specific translation. Yarowsky uses an alternative approach based on *decision lists* (Yarowsky, 1994). An ordered list of informants is built from the training corpus, the most salient informants appearing first in the list. Each informant is associated to one sense. Disambiguation of a new instance is based on the first informant in the decision list which appears in the instance context. Ng and Lee (Ng and Lee, 1996) propose an *exemplar-based approach.* The sense of an ambiguous word is determined by the instance which appears in the most similar phrase found in the training corpus. Several approaches have been compared in the *Senseval* project, a systematic evaluation of supervised techniques for word sense disambiguation (Kilgarriff, 1998; Kilgarriff and Rosenzweig, 2000).

Dictionary-based techniques work similarly as the supervised techniques but use a raw (i.e. untagged) corpus. A dictionary or a thesaurus is an additional knowledge source to define senses. In Lesk's algorithm (Lesk, 1986) the sense of an ambiguous word instance w is determined by the dictionary definition having the largest number of words in common with w context. Yarowsky proposes another approach based on the semantic categorization of the *Roget's International Thesaurus* (Yarowsky, 1992). The informants are words that often occur in the context of a semantic category of the *Roget's.*

We study here the third category of disambiguation techniques which are fully unsupervised. In such case, a particular sense cannot be assigned to an ambiguous instance. Here the problem is to automatically determine which instances can be clustered as sharing the same sense, the sense labels being arbitrary. This task can be performed through unsupervised clustering of word contexts which represent the unknown senses. Dictionary-based techniques are sometimes also referred to as unsupervised techniques since they do not require a semantically tagged corpus. To make this distinction clear, we refer to fully unsupervised disambiguation as word sense *discrimination.*

1.2 A comparative study of statistical word sense discrimination

Schütze's technique for word sense discrimination is based on a vector representation of the word contexts (Schütze, 1998). Unsupervised clustering of word senses is performed in vector space. Assuming a Gaussian distribution for each cluster, this model can be estimated with the Expectation-Maximization (EM) algorithm (Dempster *et al.* , 1977). The practical results reported by Schütze are actually based on a simplified model estimated with the K-means algorithm (Duda and Hart, 1973). This simplification was introduced for computational efficiency reasons. The first objective of this paper is to study the impact of this simplification: is there a performance gain when a real Gaussian model is estimated and, if so, at which additional computational cost?

Schütze's approach is also mentioned in (Manning and Schütze, 1999) but the probabilistic model used in this case is a discrete model of word contexts with naïve Bayes assumption. This model can also be estimated with the EM algorithm but it differs from the vector model. The discrete versus continuous nature of these models is one evidence of this distinction. The second objective of this work is to clarify this distinction and to study the relative performances of both approaches.

Statistical word sense discrimination is formally presented in section 2. Discrimination based on a discrete modeling of word contexts is described in section 3. The two variants (EM or K-means estimation) of the vector model are presented in section 4. Several experiments have been performed on the *New York Times News*. Section 5 details the corpus and our experimental protocol. Comparative results are presented in section 6.

2 Statistical word sense discrimination

In the sequel we use the following notations:

- w denotes an ambiguous word,
- $s_1, \ldots, s_K$ denote the K possible senses[1] of w,
- $c_1, \ldots, c_I$ denote the contexts of the I instances of w in a training corpus,
- $v_1, \ldots, v_J$ denote J possible informants.

Following Bayes decision theory (Duda *et al.* , 2001), word sense discrimination can be formulated as computing the sense $\hat{k}$ which maximizes the *posterior probability* $P(s_k|c)$ of sense s_k given the observed context c:

$$\hat{k} = \operatorname*{argmax}_k P(s_k|c) = \operatorname*{argmax}_k P(c|s_k)P(s_k), \tag{1}$$

where $P(c|s_k)$ is the *likelihood* of context c given the sense s_k, and $P(s_k)$ denotes the *prior* probability of sense s_k.

A discrimination model defines how the context likelihoods and prior probabilities can be computed from a set of parameters Θ. These parameters are estimated from an unlabeled training corpus, generally depending on some informants. How these parameters are estimated and which are the informants depend on the particular approach, as detailed in the following sections.

3 Naïve Bayes word sense discrimination

Given a context c_i of w, that is a window around an instance of w in the training corpus, the informants are the context words v_j. These are the content words (as opposed to stop words[2]) belonging to c_i. The context likelihood is defined as a joint probability:

$$P(c_i|s_k) = P(\{v_j \in c_i\}|s_k).$$

According to the naïve Bayes assumption, the context words are assumed to be independent[3]. In other words, the joint probability can be rewritten as

$$P(\{v_j \in c_i\}|s_k) = \prod_{v_j \in c_i} P(v_j|s_k).$$

[1] In word sense discrimination the $s_1, \ldots, s_K$ labels are arbitrary but the number K of possible senses must be decided. Automatic determination of an optimal K could also be considered.

[2] As detailed in section 5.1, stop words are conjunctions, prepositions, articles and other words, which appear often in documents yet alone may contain little meaning.

[3] This assumption is strongly arguable from a linguistic viewpoint but drastically reduces the number of parameters to be estimated and works surprisingly well in practice. Note also that, in the context of Bayes decision theory, this assumption can be reformulated in a more acceptable way as: $\text{argmax}_k\, P(s_k)P(\{v_j \in c_i\}|s_k) = \text{argmax}_k\, P(s_k)\prod_{v_j \in c_i} P(v_j|s_k)$.

The set of parameters Θ for each word w consists of the $J.K$ probabilities $P(v_j|s_k)$ and the K priors $P(s_k)$. These parameters can be estimated so as to maximize the likelihood of a training corpus. As the corpus is untagged, this is a problem of incomplete data which can be solved by the EM algorithm (Dempster *et al.* , 1977).

The EM algorithm is an iterative procedure which, starting from an initial guess Θ^0 of the parameter values, recomputes in each iteration the parameter estimates so as to increase the data likelihood or, equivalently, its log-likelihood. The log-likelihood LL of the I contexts observed in the training corpus is defined as follows[4]:

$$\begin{aligned}\mathrm{LL}(\{c_1,\ldots,c_I\}|\Theta) &= \log\prod_{i=1}^{I} P(c_i) = \sum_{i=1}^{I}\log P(c_i)\\ &= \sum_{i=1}^{I}\log\sum_{k=1}^{K} P(c_i|s_k)P(s_k)\\ &= \sum_{i=1}^{I}\log\sum_{k=1}^{K} P(s_k)\prod_{v_j\in c_i} P(v_j|s_k). \end{aligned} \tag{2}$$

In practice, the $P(v_j|s_k)$ are randomly initialized while satisfying the constraints: $\sum_{j=1}^{J} P(v_j|s_k) = 1,\ 1 \leqslant k \leqslant K$, and uniform priors are assumed: $P(s_k) = \frac{1}{K}$. The two steps of the EM algorithm are then computed iteratively as long as the log-likelihood increases.

E-step: Compute h_{ik}, an estimate of the posterior probability that sense s_k generated c_i:

$$h_{ik} = \frac{P(s_k)P(c_i|s_k)}{\sum_{l=1}^{K} P(s_l)P(c_i|s_l)} = \frac{P(s_k)\prod_{v_j\in c_i} P(v_j|s_k)}{\sum_{l=1}^{K}\Big(P(s_l)\prod_{v_j\in c_i} P(v_j|s_l)\Big)}.$$

M-step: Re-estimate $P(v_j|s_k)$ and $P(s_k)$ so as to maximize the likelihood:

$$P(v_j|s_k) = \frac{\sum_{\{c_i:v_j\in c_i\}} h_{ik}}{\sum_{j=1}^{J}\sum_{\{c_i:v_j\in c_i\}} h_{ik}},$$

where $\sum_{\{c_i:v_j\in c_i\}} h_{ik}$ sums over all contexts c_i in which v_j occurs.

$$P(s_k) = \frac{\sum_{i=1}^{I} h_{ik}}{\sum_{k=1}^{K}\sum_{i=1}^{I} h_{ik}} = \frac{\sum_{i=1}^{I} h_{ik}}{I}.$$

Once the parameters of the model have been estimated on the training corpus, the sense of a new instance of w can be assigned based on its context c. The final decision rule is:

$$\hat{k} = \operatorname*{argmax}_k P(s_k|c) = \operatorname*{argmax}_k P(s_k)\prod_{v_j\in c} P(v_j|s_k) = \operatorname*{argmax}_k\ \log P(s_k) + \sum_{v_j\in c}\log P(v_j|s_k). \tag{3}$$

[4] We follow here the presentation in (Manning and Schütze, 1999) but we use the corrected formulas as described in `http://nlp.stanford.edu/fsnlp/errata.html`.

4 Vector-based word sense discrimination

In the original Schütze's approach, words, contexts and senses are represented in a high-dimensional real-valued vector space (Schütze, 1998). Word vectors, context vectors and senses (clusters of context vectors) are presented in section 4.1. The probabilistic model and two variants of the estimation algorithm are described in sections 4.2 and 4.3.

4.1 Vector representation of senses

Word vector

A word w can be represented by a vector in which each component corresponds to a word v occurring in the corpus. The vector components represent frequencies of *co-occurrence*: the component associated with word v is the number of times that v occurs as a neighbor of w in the corpus. A neighbor is a content word occurring in a context window centered on w. These content words are the informants in this approach. For instance, if the words *legal* and *clothes* appear respectively 300 and 75 times in context windows of the word *judge*, the vector for *judge* can be represented as follows.

$$\begin{array}{rc} & \textit{judge} \\ \begin{array}{r} \\ \textit{legal} \\ \\ \textit{clothes} \\ \\ \end{array} & \begin{bmatrix} \cdots \\ 300 \\ \cdots \\ 75 \\ \cdots \end{bmatrix} \end{array}$$

Schütze examines two different ways to choose the vector dimensions: a local selection which focuses on words occurring as neighbors of the ambiguous word and ignores the rest of the corpus; a global selection which chooses the 2,000 most frequent words in the entire corpus. Moreover *word vectors* are computed only for the 20,000 most frequent words of the corpus. A 2,000-by-20,000 co-occurrence matrix can thus be derived from the corpus. To compute the most frequent words of the corpus, stop words are excluded. The best results were obtained using global selection.

Context vectors and senses

The context of an instance w is represented by a vector $\vec{x}$ obtained as the weighted sum of the *word vectors* of w neighbors (second-order co-occurrence). Given the *word vectors* $\vec{v_j}$, the *context vector* $\vec{x}$ is defined as

$$\vec{x} = \sum_{v_j \in c} a_j \vec{v_j}.$$

The weight a_j of vector $\vec{v_j}$ depends on the inverse document frequency (idf), a measure of its discriminative capability:

$$a_j = -\log \frac{d_j}{D},$$

where D denotes the number of documents in the corpus and d_j the number of documents in which v_j occurs (see section 5.1 for additional details on the corpus).

Similar *context vectors* can be seen as forming clusters in vector space. Each cluster represents one sense of an ambiguous word and can be characterized by its mean and covariance matrix. A new instance w is represented by its *context vector*. The sense of w is then assigned to the most similar cluster. Two different ways of defining the clusters are described in sections 4.2 and 4.3.

4.2 Gaussian modeling of context clusters

Context clusters are assumed to follow a Gaussian distribution. The whole model is a mixture of K Gaussian components, with one mixture component for each sense. Let $\vec{x_1}, \ldots, \vec{x_I}$ denote the context vectors ($\vec{x_i} \in \mathbb{R}^d$ is the vector associated to context c_i). $\omega_1, \ldots, \omega_K$ are the K components. Each component ω_k is characterized by some parameters: the prior probability $P(s_k)$, the mean vector $\vec{\mu_k}$ and the covariance matrix Σ_k.

Starting from an initial guess of the parameter values Θ^0, these parameters are reestimated with the EM algorithm so as to maximize the training data likelihood. The initialization procedure typically follows from a hard clustering of the context vectors as detailed in section 4.3. Such clustering defines a first estimate of the K mean vectors. The context vectors are then assigned to their closest mean and the cluster covariance matrices can be computed. The initial prior of cluster ω_k is defined as $P(s_k) = \frac{i_k}{\sum_{k=1}^{K} i_k} = \frac{i_k}{I}$ where i_k is the number of vectors assigned to cluster ω_k and I is the total number of context vectors.

The log-likelihood of the I contexts observed in the training corpus is defined as

$$\mathrm{LL}(\{\vec{x_1}, \ldots, \vec{x_I}\}|\Theta) = \log \prod_{i=1}^{I} P(\vec{x_i}) = \sum_{i=1}^{I} \log \sum_{k=1}^{K} P(s_k) f_k(\vec{x_i}), \tag{4}$$

where $f_k(\vec{x_i})$ denotes the value of the Gaussian density in $\vec{x_i}$

$$f_k(\vec{x_i}) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_k|}} \exp\left[-\frac{1}{2}(\vec{x_i} - \vec{\mu_k})^T \Sigma_k^{-1} (\vec{x_i} - \vec{\mu_k})\right].$$

The two steps of the EM algorithm are then computed iteratively as long as the log-likelihood increases.

E-step: From the parameter values at iteration r, compute h_{ik}, the posterior probability that ω_k generated $\vec{x_i}$:

$$h_{ik} = \frac{P(s_k) f_k(\vec{x_i})}{\sum_{l=1}^{K} P(s_l) f_l(\vec{x_i})}.$$

M-step: Re-estimate the parameters at iteration $r+1$ so as to maximize the likelihood:

$$\begin{aligned}
\vec{\mu_k}^{r+1} &= \frac{\sum_{i=1}^{I} h_{ik} \vec{x_i}}{\sum_{i=1}^{I} h_{ik}}, \\
\Sigma_k^{r+1} &= \frac{\sum_{i=1}^{I} h_{ik} (\vec{x_i} - \vec{\mu_k}^r)^T (\vec{x_i} - \vec{\mu_k}^r)}{\sum_{i=1}^{I} h_{ik}}, \\
P^{r+1}(s_k) &= \frac{\sum_{i=1}^{I} h_{ik}}{\sum_{l=1}^{K} \sum_{i=1}^{I} h_{il}} = \frac{\sum_{i=1}^{I} h_{ik}}{I}.
\end{aligned}$$

Once the parameters have been estimated on the training corpus, the sense of a new instance of w can be assigned from the vector $\vec{x}$ associated to its context c. The final decision rule is:

$$\hat{k} = \underset{k}{\operatorname{argmax}}\, P(s_k)P(c|s_k) = \underset{k}{\operatorname{argmax}}\, P(s_k)f_k(\vec{x}). \tag{5}$$

4.3 Hard clustering of context vectors

The probabilistic model described in section 4.2 defines a Gaussian mixture of K components. Any context vector $\vec{x}$ can be seen as being generated by all K components. This approach is sometimes called soft-clustering since a vector is not deterministically assigned to a particular cluster (i.e. a mixture component). An alternative approach is hard clustering where $\vec{x}$ is assigned to its closest cluster mean according to the euclidean distance in vector space. Hard clustering can either be used as initialization before reestimation of a Gaussian model or as a sense discrimination technique as such.

Hard clustering can be performed in two steps with group-average agglomerative clustering (GAAC) and K-means (Schütze, 1998). GAAC is a bottom-up hierarchical clustering algorithm. Starting from a randomly selected subset of the I context vectors, GAAC iteratively agglomerates vectors into K clusters by merging most similar vectors first. The similarity measure used is the cosine:

$$sim(\vec{x},\vec{y}) = cos(\vec{x},\vec{y}) = \frac{\vec{x}\cdot\vec{y}}{|\vec{x}||\vec{y}|},$$

which simply amounts to the dot product for normalized vectors. The computational complexity of GAAC is $\mathcal{O}(n^2)$ where n denotes the size of the initial vector subset. In practice, a subset of $\sqrt{I}$ vectors can be selected. This allows to compute reasonably good cluster means in $\mathcal{O}(I)$.

The K cluster means serve as initialization for the K-means algorithm which runs in $\mathcal{O}(I)$ (Duda *et al.* , 2001). The I context vectors are first assigned to their closest means. Cluster means $\vec{\mu_k}$ are then recomputed. This process is iterated as long as the $\vec{\mu_k}$ vectors change.

Once the parameters have been estimated on the training corpus, the sense of a new instance of w can be assigned from the vector $\vec{x}$ associated to its context c by finding its closest mean. The final decision rule is:

$$\hat{k} = \underset{k}{\operatorname{argmin}} \parallel \vec{x} - \vec{\mu_k} \parallel . \tag{6}$$

Equation (6) is equivalent to equation (5) provided the K senses are assumed equally likely ($P(s_k) = \frac{1}{K}$) and a common covariance matrix is assumed for all K mixture components. This property illustrates that the model presented here is a simplified version of the Gaussian mixture model presented in section 4.2. Note however that the K estimated means are not necessarily the same in both models.

5 Experimental Assessment

Section 5.1 describes the corpus used in our experiments. The role of pseudowords and how they are used is described in section 5.2. Other details of the experimental protocol are presented in section 5.3. As detailed in the sequel, we follow here as much as possible Schütze's choices in parameter setting for comparison purposes.

5.1 Corpus and stop list

The available corpus selected for our experiments is the *New York Times News* of 1997. The training set comes from the first six months issues (January 1997 till June 1997). It contains 74,847,796 word tokens ($\sim$ 500 megabytes). There are 485,936 different words in this set. The words are not stemmed: singular and plural forms of a same word count for two different words and each form of a same verb counts for a different word.

This corpus is divided into *documents*. Each document corresponds to an article in the newspaper. The training set is made of 116,010 documents. The mean number of words per document is 645 with a standard deviation of 392.

The test set is extracted from the first 17 days of December 1997. The test set contains 7,857,354 word tokens ($\sim$ 50 megabytes) among which 135,502 different words. The mean number of words per document is 621 and the standard deviation is 397.

The proportion between training and test set have been chosen so as to represent a similar amount of data[5] as in Schütze's experiments (Schütze, 1998). The same context window size (50 tokens) have been chosen as well. The context window of an instance of w is made of up to 25 tokens on the left and 25 tokens on the right of w. A context window never crosses the limit of a document and only content words are considered inside it. Content words are defined as any word not belonging to a stop list. Our stop list is made of 574 stop words as defined in `http://lingo.lancs.ac.uk/devotedto/corpora/software.htm`[6]. Stop words are conjunctions, prepositions, articles and other words, which appear often in documents yet alone may contain little meaning.

5.2 Pseudowords

Ambiguous word	Sense	Distribution	**Pseudoword**	Senses	Training	Test
accident	chance	14%	banana-moon	banana	452	39
	crash	86%		moon	2,452	263
				Total	2,904	302
motion	physical movement	39%	animal-river	animal	2,389	219
	proposal for action	61%		river	6,104	362
				Total	8,493	581
train	to teach	30%	rely-illustration	rely	1,669	149
	line of railroad cars	70%		illustration	3,541	334
				Total	5,210	483
interest	feeling of special attention	31%	data-school	data	9,154	1,032
	charge on borrowed money	69%		school	29,095	2,468
				Total	38,249	3,500
suit	set of garments	12%	railway-admission	railway	550	27
	action or process in a court	88%		admission	1,974	189
				Total	2,524	216

Table 1: *Pseudoword frequencies.*

[5] Schütze used the *New York Times News* of 1989-90. His training and test sets contain respectively 60.5 million word tokens and 5.4 million word tokens.

[6] Our stop list is the union of 4 stop lists found under the reference *Function Words/Stop Lists for English*.

In order to test the performance of sense discrimination algorithms on naturally ambiguous words, a large number of instances have to be disambiguated by hand. As this is a time-consuming task, it is convenient to generate artificially ambiguous words: *pseudowords*. A pseudoword is the concatenation of two or more natural words.

Discrimination of pseudowords does not exactly reflect the discrimination task of real ambiguous words but precautions can be taken so as to best reflect a natural case (Gaustad, 2001). For example, the real ambiguous word *accident* has two main senses: *crash* and *chance*. A hundred instances of *accident* were manually tagged to determine its sense distribution. The corpus is then searched for two unambiguous words having a frequency of occurrence roughly fitting the ambiguous word sense distribution. In the case of *accident*, the unambiguous words *banana* and *moon* satisfy this requirement. All instances of *banana* and *moon* in the training corpus are then replaced by the pseudoword *banana-moon*. Table 1 gives the pseudowords built for five natural ambiguous words (with their respective sense distribution) and their frequencies of occurrence in the training and test sets.

5.3 Experimental protocol

All discrimination models include some random initialization before reestimation. In the naïve Bayes discrimination model (section 3), the likelihoods $P(v_j|s_k)$ are initialized at random. In the hard clustering discrimination model (section 4.3) the K mean vectors derive from a randomly selected subset of the I context vectors. The result of the estimated K means is also used to initialize a Gaussian model (section 4.2). As the EM algorithm is only guaranteed to find a local optimum of the likelihood, its performance depends indirectly on this initialization. Hence all experiments are repeated 10 times while varying the random seeds. Averaged results over these 10 independent runs are reported in section 6. In all experiments so far, the value of K is equal to 2 (binary sense discrimination).

The implementation used for the Gaussian model assumes a diagonal covariance matrix for each cluster. Possible correlations between the components of the context vectors are ignored but the number of parameters to be estimated for each ambiguous word is reduced to $K(1+2d)$, where d denotes the dimension of the vector space. In all cases, the result of the hard clustering techniques (the K means representing Kd parameters) was used to initialize the Gaussian model. Hence we were able to check whether the Gaussian model further improves the performance obtained with hard clustering. In all tests of the vector model a global selection of the vector dimensions was chosen (see section 4.1).

Pseudoword	**Occurrences (I)**	**Context words (J)**	**Parameters**
banana-moon	2,904	11,449	22,900
animal-river	8,493	27,413	54,828
rely-illustration	5,210	17,763	35,528
data-school	38,249	56,008	112,018
railway-admission	2,524	11,593	23,186
Average	11,512	24,845	49,692

Table 2: *Number of occurrences and informants in the local naïve Bayes approach.*

In the naïve Bayes approach (section 3), the J informants to discriminate the senses of an ambiguous instance w are the J content words belonging to context windows around w in the training corpus. This implies that the number J and identity of informants depend on the word

w. We refer to this approach as *local naïve Bayes*. The total number of parameters of the local model is $K(1 + J)$. Table 2 reports the number of informants for each pseudoword and the corresponding number of parameters.

An alternative approach is to consider the same set of informants for all ambiguous words. In this case the 20,000 most frequent content words in the training corpus are considered. The number of parameters (40,002) does no longer depend on the word to disambiguate. We refer to this approach as *global naïve Bayes*.

6 Results

Table 3 gives the discrimination results for the pseudowords considered in these experiments. The first two measures (S1, S2) for each pseudoword give the percentage of correct senses for each of the two words making the pseudoword. As the sense labels are arbitrary in a sense discrimination experiment, the most frequent sense (S1) is considered to be attributed to the most frequent word in the training (e.g. *moon* for the *banana-moon* pseudoword).The accuracy gives the total proportion of correctly labeled instances for both senses. In each case, the mean (μ) and standard deviation (σ) of the performances obtained over 10 independent runs are reported. The last line reports the average accuracy (mean and standard deviation) obtained for the five pseudowords.

		Naïve Bayes				**Vector Model**			
Pseudowords		**Local**		**Global**		**K-Means**		**Gaussian**	
		μ	σ	μ	σ	μ	σ	μ	σ
banana-moon	S1	56.9	2.4	58.1	2.0	68.4	0.0	100.0	0.0
	S2	34.4	17.7	27.2	12.3	0.0	0.0	0.0	0.0
	accuracy	54.0	2.8	54.1	1.5	59.6	0.0	87.1	0.0
animal-river	S1	66.4	14.2	76.3	15.7	73.8	13.1	84.1	0.4
	S2	34.4	17.8	29.5	14.6	20.8	7.0	7.5	0.2
	accuracy	54.3	7.0	58.7	6.4	53.8	5.5	55.2	0.2
illustration-rely	S1	88.7	5.9	87.9	7.2	81.9	5.9	99.9	0.2
	S2	35.2	11.2	31.4	10.3	55.5	39.7	0.5	0.7
	accuracy	72.2	3.9	70.5	5.9	73.8	8.8	69.2	0.1
data-school	S1	78.2	16.0	92.3	9.9	83.1	0.9	93.5	0.2
	S2	40.0	34.5	59.0	16.3	58.8	17.0	44.4	15.0
	accuracy	66.9	14.7	82.5	4.5	75.9	4.4	79.0	0.3
railway-admission	S1	76.8	12.7	79.3	9.9	56.2	15.4	99.5	0.2
	S2	32.9	23.5	48.5	21.1	67.0	22.3	13.3	4.7
	accuracy	71.3	11.2	75.5	6.7	57.5	10.7	88.8	0.4
Average accuracy		63.7	7.9	68.3	5.0	64.1	5.9	75.9	0.2

Table 3: *Discrimination results.*

The average accuracy illustrates that the two reference models proposed respectively in (Manning and Schütze, 1999) and (Schütze, 1998), namely the local naïve Bayes and K-means vector model, perform roughly as well. Note however that for a given pseudoword these approaches can give significantly different results. For instance, the K-means vector model wrongly attributes all instances of *banana* to the *moon* cluster. Moreover this result is not affected by the random initialization as it does not change over the 10 independent runs ($\sigma = 0$). In contrast, the local naïve Bayes model splits test instances between the 2 senses.

The global naïve Bayes model slightly improves over the reference models. The Gaussian vector model performs significantly better on average than the reference models. Moreover the variance of the results is also decreased showing that this approach is less sensitive to a particular initialization. Note that the Gaussian model tends to favor the majority sense in several cases.

Table 4 summarizes the computational cost for estimating[7] the discrimination models and the number of estimated parameters[8] in each case. As the number of occurrences of the pseudowords varies in the training set (see table 2), the reported CPU times[9] correspond to the estimated times for processing 3,000 occurrences in all cases. This analysis can probably be refined with a detailed profiling and optimization of the programming code but it illustrates already the tractability of all approaches considered so far. The Gaussian model offers the best accuracy and is parsimonious as it has 5 times less parameters than the global naïve Bayes model. The average number of iterations required to converge is reported in the last column.

Method	**Accuracy**	**CPU Time (sec)**	**Number of parameters**	**Iterations**
Naïve Bayes (local)	63.7	.4	49,692	12
Naïve Bayes (global)	68.3	.5	40,002	17
Vector Model (K-means)	64.1	10.5	4,000	10
Vector Model (Gaussian)	75.9	36.7	8,002	6

Table 4: *Accuracy/CPU Time trade-off.*

7 Conclusion and future work

We compared in this work several word sense discrimination techniques. The vector model proposed by Schütze (Schütze, 1998) can significantly be improved when a real Gaussian model is estimated instead of its hard clustering approximation. This performance gain is obtained with an additional computational cost but the estimation procedure remains very efficient in all cases. The Gaussian model tested here includes a diagonal covariance matrix for each sense. We could also consider a full covariance matrix but this would significantly increase the number of parameters and the computation time. This option will be evaluated in further experiments.

The naïve Bayes model described in (Manning and Schütze, 1999) has also been implemented and its average performance is comparable with the hard clustering approach. Our experiments demonstrate that a performance gain is obtained when the same context informants are used for all pseudowords. This global approach has the advantage of a common set of parameters for all ambiguous words which are more reliably estimated over the whole training corpus.

Our results are not fully comparable with Schütze's experiments even though we followed the same experimental protocol, as closely as possible. The first reason is that the used corpora differ (New York Times News 97 versus 89-90) but a similar amount of data was used. The stop lists differ (574 words versus 930 words). The pseudowords were also built in a slightly different way. We argue that our pseudoword design better reflects the discrimination task for naturally ambiguous words while not requiring time consuming labeling of the corpus. Schütze also demonstrated the advantage of reducing the vector space dimension with *Singular Value Decomposition* (SVD) (Berry, 1992). Including SVD in the vector model is our very next task.

[7] The figure reported corresponds to the time for estimating the discrimination models from precomputed context vectors or context windows. Hence this time does not include the preprocessing of the corpus to extract the contexts and filter out the stop words.

[8] For the local naïve Bayes model, the average number of parameters has been reported (see table 2).

[9] The CPU times are measured on a laptop with a 600 MHz processor and 384 Mb of RAM. All estimation programs are written in C.

Several additional options and extensions will be considered in the future. In particular we will study:

– the influence of the context window size (currently 50 words around the ambiguous instance); we expect that this size can be significantly reduced,
– the influence of stemming and the definition of the stop list,
– the number K of senses considered (currently only binary sense discrimination is considered), and the automatic determination of an optimal K value,
– the dimension of the original vector space and the final space dimension after singular value decomposition,
– smoothing techniques to improve estimates of the global naïve Bayes model.

References

Berry, M. W. (1992). Large-scale sparse singular value decomposition. *The International Journal of Supercomputer Applications*, **6**(1), 13–49.

Brown, P., Della Pietra, S., Della Pietra, V., and Mercer, R. (1991). Word-sense disambiguation using statistical method. *ACL*, **29**, 139–145.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum Likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society Ser. B (methodological)*, **39**, 1–38.

Duda, R., and Hart, P. (1973). *Pattern Classification and Scene Analysis.* New York: Wiley.

Duda, R., Hart, P., and Stork, D. (2001). *Pattern Classification.* New York: Wiley.

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

Gaustad, T. (2001). Statistical Corpus-Based Word Sense Disambiguation: Pseudowords vs Real Ambiguous Words. *Companion Volume to the Proc. of the 39th Annual Meeting of the Association for Computational Linguistics (ACL/EACL 2001) – Proceedings ot the Student Research Workshop.*

Kilgarriff, A. (1998). Gold Standard Datasets for Evaluating Word Sense Disambiguation Programs. *Computer Speech and Language*, **12**(4), 453 – 472.

Kilgarriff, A., and Rosenzweig, J. (2000). English Senseval : Report and Results. *Proc. of the Second International Conference on Language Resources and Evaluation*, pages 1239–1244.

Lesk, M. (1986). Automatic sense disambiguation : How to tell a pine cone from an ice cream cone. *Proc. of the 1986 SIGDOC Conference.* Association for Computing Machinery, New-York, pages 24–26.

Manning, C.D., and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing.* Cambridge, London: MIT Press.

Ng, H., and Lee, H. (1996). Integrating mutliple knowledge sources to disambiguate word sense : an exemplar-based approach. *Proc. of 34th Annual Meeting of the Society for Computational Linguistics*, pages 40–47.

Schütze, H. (1998). Automatic Word Sense Discrimination. *Computational Linguistics*, **24**, 97–124.

Yarowsky, D. (1992). Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. *Proc. of the 33th Annual Meeting of the Association for Computational Linguistics*, pages 189–196.

Yarowsky, D. (1994). Decision Lists for Lexical Ambiguity Resolution : Application to Accent Restoration in Spanish and French. *Proc. of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 88–95.