Local Vector-based Models for Sense Discrimination

M.-C. de Marneffe, C. Archambeau, P. Dupont, M. Verleysen*

Machine Learning Group, Université catholique de Louvain (UCL)
Place du Levant 3, B-1348 Louvain-la-Neuve, Belgium
{mcdm,pdupont}@info.ucl.ac.be, {archambeau,verleysen}@dice.ucl.ac.be
www.ucl.ac.be/mlg

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 high dimensional vector representation of word senses was proposed by Schütze [10]. While this model was assumed to be Gaussian, results were only reported for the K-means approximation. In this work, a local vector-based model of reduced dimensionality which is linguistically coherent and can be computed for multivariate Gaussian mixtures is proposed. Several practical experiments are conducted on the New York Times News 1997 corpus. They show the advantages of unrestricted Gaussian models compared to K-means. The correct discrimination rate is further increased when using regularized Gaussian models as proposed in [2].

1 Introduction

The purpose of word sense disambiguation is to determine the exact sense of an instance of an ambiguous word used in a certain context. Disambiguation is potentially useful in any linguistic application for which word sense matters as in automatic translation, text categorization, speech understanding, etc.

Word sense disambiguation techniques can be divided into three broad categories: supervised techniques, dictionary (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*. The difference between them lies in the knowledge sources used (e.g. dictionary).

[*] M.V. is a Senior Research Associate of the Belgian National Fund for Scientific Research (FNRS).

Sixth International Workshop on Computational Semantics IWCS'05, Tilburg, The Netherlands, January 12-14, pp. 163-174, 2005.

Supervised techniques require a semantically tagged corpus, i.e. a set of documents in which each ambiguous instance w is correctly labeled with a semantic tag, which serves as training corpus. The *senses* are defined by the semantic tags present in the corpus. The *contexts* consist of windows around instances of w and *informants* are the words belonging to those context windows. For dictionary-based techniques, a raw (i.e. untagged) corpus is used but a dictionary or a thesaurus gives the additional knowledge to define senses. The third technique, which is studied here, is fully unsupervised and only needs a raw corpus. 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. Note that 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.*

Due to the small size of semantically tagged corpora and as discrimination only requires raw data which is easy to obtain, it is worthwhile to concentrate effort on discrimination techniques. Schütze introduced vector representation of word contexts where unsupervised clustering of word senses is performed in a high dimensional vector space [10]. Assuming a Gaussian distribution for each cluster, this model can be estimated iteratively with the expectation-maximization (EM) algorithm [4]. For computational reasons, Schütze used a simplified model estimated with the K-means algorithm [5]. However the use of the Gaussian model improves the results as shown in [3]. Unfortunately, in a high dimensional vector space, a diagonal covariance matrix has to be assumed to simplify the computation and avoid numerical difficulties. In this paper, following the ideas of Sinclair and Véronis about word senses [11, 12], an adapted version of Schütze's technique is proposed to compute word contexts in a smaller dimensional space. This approach is linguistically more coherent and is able to avoid some simplification assumptions of the Gaussian model. The algorithm which is then used to compute the model is a regularized version of the EM algorithm for mixture models proposed in [2]: the modified M-step uses the regularized Mahalanobis distance to determine the shape of the Gaussian components.

A short overview of the theories of Sinclair and Véronis, which bring us to adapt Schütze's representation, is given in section 2. Schütze's model and our proposed model are presented in section 3. The improved EM algorithm used in the experiments is detailed in section 4. Several experiments have been performed on the *New York Times News*; they are described in section 5 as well as the results obtained.

2 Distributional information

In the task of disambiguation, the important question to solve is the following: given an instance of an ambiguous word, what does its meaning indicate? Sinclair claims that "the structure realizes sense and therefore normally differentiates one sense from another" [11]. By *structure*, Sinclair means collocations (i.e. words occurring with each other) and similar patterns. This point of view was already the one of Meillet: "the sense of a word is defined only by the average of its linguistic uses" [8]. Surface clues, i.e. syntax and collocations, are thus valuable informants for meaning. According to Sinclair and Véronis, a ten words window around (i.e. five before and five after) the ambiguous word gives enough context.

Véronis summarizes this idea by the term *distributional information* which gathers syntactic and collocational information. He gives an example of the power of such information using the ambiguous French word *barrage* (= dam, roadblock, opposition) [12]. In the two following sentences, the prepositions *sur* (on) or *à* (to), which are syntactic information, allows us to disambiguate *barrage*:

(1) *le barrage sur le Rhône*, *le barrage sur l'autoroute* (the dam on the Rhône river, the roadblock on the highway)

(2) *le barrage à la loi* (the opposition to the law)

The collocation information gives enough clues to differentiate the "dam" or "roadblock" senses of *barrage. Barrage* (= dam) will frequently occur with verbs such as *édifier* (edify), *construire* (build), *démolir* (destroy), while *barrage* (= roadblock) will be found with verbs such as *dresser* (put up), *franchir* (cross), *démanteler* (dismantle).

As Véronis points out, ordinary dictionaries are inappropriate for the disambiguation task because there are chiefly concerned with the definition of meaning and not with distributional information. This is one of the reasons this work focuses on discrimination techniques and try to adapt the model proposed by Schütze in order to add distributional information.

3 Local context representation following Schütze's model

First, the model as proposed by Schütze is presented. Then, the way to adapt the model according to linguistic facts in order to work in vector spaces of smaller dimensionality is described.

3.1 Schütze approach

In the original Schütze's approach, words, contexts and senses are represented in a high-dimensional real-valued vector space [10]. Two types of representation can be distinguished: word vectors and context vectors.

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.

Schütze examines two different ways to choose the vector dimension: a local selection which focuses on the 1,000 most frequent 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. In the global manner, *word vectors* are computed only for the 20,000 most frequent words of the corpus. To compute the most frequent words of the corpus, stop words are excluded. Stop words are conjunctions, prepositions, articles and other words, which appear often in documents yet alone may contain little meaning. In our adapted model described in section 3.2, some stop words are however used as they may contain important clues for sense discrimination.

Context vectors and senses

The context of an instance w is represented by a vector $\mathbf{x}$ obtained as the weighted sum of the *word vectors* of w's neighbors. Given a *context* C and *word vectors* $\mathbf{v}_i$ of the corresponding neighbors v_i of w, the *context vector* $\mathbf{x}$ is defined as follows:

$$\mathbf{x} = \sum_{v_i \in C} a_i \mathbf{v}_i. \tag{1}$$

The weight a_i of vector $\mathbf{v}_i$ is the inverse document frequency, a measure of its discriminative capability: $a_i = -\log \frac{d_i}{D}$, D denoting the number of documents in the corpus and d_i the number of documents in which v_i occurs.

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. The sense of a new instance w is then assigned to the most similar cluster.

3.2 Adapted model

As collocations matter in disambiguation, local selection is used. The corpus has been lemmatized[1], which was not the case in Schütze's model. For each ambiguous word, *all* the lemmas occurring in a small lemmas window around the ambiguous word instances in the corpus are taken into account. These lemmas are the informants and their number vary according to the ambiguous word. The word vector for each informant is computed in the same way as Schütze did, except that the vector dimension is drastically reduced.

A stop list (which normally contains stop words) is used while computing the most frequent lemmas, but it does not include prepositions given their importance in the disambiguation process [12], as stressed in section 2.

Focusing on local selection, it is now the concept of *contexts* around an ambiguous word which makes more sense than the concept of *documents* in which the ambiguous word occurs. The weight computation is thus slightly different: D denotes the number of contexts (i.e. the window of lemmas) examined for an ambiguous word in the corpus and d_i the number of contexts in which v_i occurs.

4 Unsupervised discrimination

In this section, K-means, unrestricted Gaussian mixture and the regularized EM for Gaussian mixtures are presented.

4.1 K-means

K-means [5] is also referred to as hard clustering because each vector $\mathbf{x}$ is assigned to its closest cluster mean (or center) $\mathbf{c}_j$

[1] The part-of-speech tagger *TreeTagger* which is also a lemmatizer (links every word to its lexical entry in a dictionary) and achieves 96.36% accuracy is used, cfr [9].

according to the Euclidean distance in vector space: $\Delta_E = (\mathbf{x} - \mathbf{c}_j)^T (\mathbf{x} - \mathbf{c}_j)$. Given K cluster means, the context vectors of the training set are first assigned to their closest means. Cluster means $\mathbf{c}_j$ are then recomputed. This process is iterated as long as the $\mathbf{c}_j$ 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 $\mathbf{x}$ associated to it by finding its closest mean.

4.2 Finite Gaussian mixtures

Mixtures of Gaussian distributions are commonly used for unsupervised learning tasks [7]. They have been shown to be effective in numerous clustering problems ranging from fire detection to tissue segmentation of brain magnetic resonance images. Provided the number of components in the mixture is known, the maximum likelihood estimates of the model parameters can be computed in an elegant way by applying the expectation-maximization (EM) algorithm [4].

A finite Gaussian mixture is defined as a linear combination of K Gaussian component densities:

$$p(\mathbf{x}) = \sum_{j=1}^{K} P(j) p(\mathbf{x}|j). \tag{2}$$

The mixing proportions $P(j)$ are non-negative and must sum to one, and $p(\mathbf{x}|j)$ is the distribution of cluster j. It is defined by

$$p(\mathbf{x}|j) = (2\pi)^{-\frac{d}{2}} |\Sigma_j|^{-1/2} \exp\left(-\Delta/2\right), \tag{3}$$

where $\Delta = (\mathbf{x} - \mathbf{c}_j)^T \Sigma_j^{-1} (\mathbf{x} - \mathbf{c}_j)$ is the Mahalanobis distance and d the dimension of the context vectors. Fitting the mixture components to the clusters consists then in estimating the centers $\mathbf{c}_j$, the covariance matrices Σ_j and the mixing proportions $P(j)$ based on the observed data $\{\mathbf{x}_n\}_{n=1}^{N}$.

In order to maximize the likelihood $L = \prod_{n=1}^{N} p(\mathbf{x}_n)$ of the observed data, EM operates in two stages. First, in the E-step, the expected value of some "unobserved" data is computed, using the current parameter estimates and the observed data. The "unobserved" data are called responsibilities and they represent the probability that a data point was generated by a well-defined mixture component. Subsequently, during the M-step, the expected values computed in the E-step are used to update the model parameters in such a way that the likelihood is increased. Each iteration step t can be summarized as follows [7]:
E-step:

$$P^{(t)}(j|\mathbf{x}_n) = \frac{p^{(t)}(\mathbf{x}_n|j)P^{(t)}(j)}{p^{(t)}(\mathbf{x}_n)}. \tag{4}$$

M-step:

$$\mathbf{c}_j^{(t+1)} = \frac{\sum_{n=1}^{N} P^{(t)}(j|\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} P^{(t)}(j|\mathbf{x}_n)}, \tag{5}$$

$$\Sigma_j^{(t+1)} = \frac{\sum_{n=1}^{N} P^{(t)}(j|\mathbf{x}_n)\left(\mathbf{x}_n - \mathbf{c}_j^{(t+1)}\right)\left(\mathbf{x}_n - \mathbf{c}_j^{(t+1)}\right)^T}{\sum_{n=1}^{N} P^{(t)}(j|\mathbf{x}_n)}, \tag{6}$$

$$P^{(t+1)}(j) = \frac{1}{N}\sum_{n=1}^{N} P^{(t)}(j|\mathbf{x}_n). \tag{7}$$

Once the parameters have been estimated on the training corpus, the sense of a new instance of w can be assigned from the vector $\mathbf{x}$ associated to it by applying Bayes' rule.

4.3 The regularized Mahalanobis distance

Numerical difficulties in the use of EM for Gaussian mixtures may arise due to singularities in the likelihood function, leading to a component to collapse (i.e. its width tends to zero). This is even more troublesome in high dimensional spaces or when the data set is sparse. This problem was discussed in [1] and accredited to the concept of isolation. In [2], it was proposed to use the regularized Mahalanobis distance in order to improve the estimation of the clusters.

The multivariate Gaussian uses the Mahalanobis distance Δ to determine its shape. When the number of data samples contributing to the computation of the covariance matrix of a component is small with respect to the dimension d of the data samples, this matrix may be singular. Moreover, the use of Δ tends to produce hyperellipsoidal components, leading to unusually large and elongated densities. By contrast, when one considers the Euclidean distance Δ_E, large data clusters tend to split unnecessarily, as the component densities are hyperspherical.

Based on the hyperspherical character of Δ_E and the hyperellipsoidal character of Δ, one can construct a regularized Mahalanobis distance Δ_R as a convex combination of both distances:

$$\Delta_R = (1-\lambda)\Delta + \lambda\Delta_E, \lambda \in [0,1]. \tag{8}$$

Parameter λ controls the trade-off between hyperspherical and hyperellipsoidal components. Large value of λ should be used when the covariance matrices cannot be estimated reliably.

Consider again the E- and M-step. Introducing the regularized Mahalanobis distance consists in adapting, at each iteration step t, the covariance matrix of each component density according to (8). Therefore, the following adaptation rule is inserted at the end of the M-step:

$$\Sigma_j^{(t+1)} = \left[(1-\lambda)\left(\Sigma_j^{(t+1)} + \epsilon I\right)^{-1} + \lambda\tau I\right]^{-1}, \qquad (9)$$

where I is the $d \times d$ identity matrix and ϵ is called the safety factor. Its role is to stabilize the learning process whenever needed by converting a singular matrix to a non-singular one. Using different values of ϵ do not make much difference as long as they are significantly smaller than the variance of the data samples [2]. Finally, τ is a scaling factor taking the range of the data into account. It is computed according to the rule-of-thumb $\tau = \sqrt[d]{K}/\sigma_{\mathbf{x}}^2$, where $\sigma_{\mathbf{x}}$ is the standard deviation of the data.

5 Experiments and results

The role of pseudowords is described in section 5.1. Section 5.2 describes the corpus used in the experiments. Results of the different experiments are given in section 5.3.

5.1 Pseudowords

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 union 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 [6]. For example, the real ambiguous word *motion* has two main senses: *physical movement* and *proposal for action.* A hundred instances of *motion* 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 *motion*, the unambiguous words *animal* and *river* satisfy this requirement. All instances of *animal* and *river* in the training corpus are then replaced by the pseudoword *animal-river.*

The first four rows of table 1 give the pseudowords built to represent four natural ambiguous words (respectively motion, train, interest and suit) and their frequencies of occurrence in

Pseudoword	# lemmas	Senses	Training	Test
animal-river	10, 932	animal	1, 662	156
		river	4, 103	270
rely-illustration	5, 859	rely	1, 073	109
		illustration	2, 458	243
data-school	22, 908	data	5, 855	707
		school	20, 880	1, 955
railway-admission	3, 968	railway	354	23
		admission	1, 240	116
railway-train	4, 911	railway	354	23
		train	1, 437	150
jury-judge	12, 149	jury	4, 191	380
		judge	6, 924	922
lawyer-judge	15, 274	lawyer	5, 454	668
		judge	6, 924	921

Table 1: *Number of pseudoword occurrences and number of lemmas in the contexts.*

the training and test sets. For the components of the last three pseudowords, nouns which belong to the same field and should occur in quite similar semantic contexts have been chosen.

5.2 Corpus

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 74,369,799 word ($\sim$ 500 megabytes). The test set is extracted from the first 17 days of December 1997. The test set contains 7,805,395 word ($\sim$ 50 megabytes).

Contexts of lemmas (i.e. lemmas window around the ambiguous word) are considered for each ambiguous word. Table 1 gives the number of lemmas occurring in the contexts and not belonging to the stop list for each pseudoword used as ambiguous word in our experiments. A small window size is used: five lemmas before and after the ambiguous word.

5.3 Results

Vectors of dimension 20 corresponding to the 20 most frequent lemmas are used. Experiments have also been done with 5, 10 and 50 dimension vectors. Five and ten dimensions do not provide enough information for disambiguation while fifty dimension did not improve the results.

Experiments have been run with the K-means algorithm and with the EM algorithm for Gaussian mixtures. For the latter, three implementations have been used: one assuming a diagonal covariance matrix for each cluster (*dFGM* in table 2), one using a full covariance matrix (*FGM*) and one using the regularized Mahanalobis distance as explained in section 4.3 (*mFGM*).

All algorithms require an a priori fixed number of clusters. In the experiments, the number of clusters is fixed at two (binary sense discrimination). Table 2 gives the discrimination results for the pseudowords considered. The first two measures (S1, S2) gives 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. *river* for the *animal-river* pseudoword). The accuracy is taken as the average correct discrimination rate for both senses S1 and S2; clusters are thus considered equiprobable, disregarding the number of word vectors in each cluster. The accuracy given here differs from the one in [10] and [3] where it was computed as the total percentage of correct discrimination, disregarding the sense labels. However we believe that the former reflects better the quality of the disambiguation process in general. The optimal value for the parameter λ is selected by exhaustive search, its range being known ($\lambda \in [0, 1]$).

The accuracy of the Gaussian model is in general higher than the accuracy of K-means. Working with a full covariance matrix is thus worthy and, in all cases, the regularized FGM gives better results. Remark that using unrestricted FGM becomes feasible in a local context as the number of parameters to estimate are severely reduced. Indeed, in the local vector-based approach the dimensionality of the vector space is reduced by a factor 100 as compared to the global model which was shown to performed best in Schütze's experiments. Remark also that the computational overhead of regularized FGM is negligible compared to regular FGM, while improving discrimination. A performance of roughly 60-70% is achieved for most pseudowords. The good performance for *rely-illustration* is probably due to the fact that the two components making the pseudoword have different parts of speech (noun, verb). For the first four pseudowords in the table [2], the results are on average better (accuracy of 67.1%, 68.7% and 71.7% for the diagonal FGM, FGM and the regularized FGM re-

[2] Average results on these four pseudowords are reported for comparison purposes.

Pseudowords		**K-means**	**dFGM**	**FGM**	**mFGM**
animal-river	S1	77.4	52.6	48.9	48.9
	S2	15.4	57.7	78.9	78.9
	acc.	46.4	55.1	63.9	$(\lambda = 0.0)$ 63.9
rely-illustration	S1	95.1	90.5	92.5	90.5
	S2	67.0	94.5	89.0	100.0
	acc.	81.0	92.5	90.8	$(\lambda = 0.4)$ 95.3
data-school	S1	84.1	65.9	59.2	78.1
	S2	19.1	50.3	48.5	40.3
	acc.	51.6	58.1	53.9	$(\lambda = 1.0)$ 59.2
railway-admission	S1	92.4	47.4	58.6	45.7
	S2	25.6	78.3	73.9	91.3
	acc.	59.0	62.8	66.3	$(\lambda = 0.2)$ 68.5
average accuracy		59.5	67.1	68.7	71.7
railway-train	S1	89.3	56.0	88.0	92.0
	S2	41.7	60.9	52.2	52.2
	acc.	65.5	58.4	70.1	$(\lambda = 0.2)$ 72.1
jury-judge	S1	85.7	52.9	53.7	56.8
	S2	24.2	52.1	59.7	58.1
	acc.	54.9	52.5	56.7	$(\lambda = 0.1)$ 57.0
lawyer-judge	S1	85.7	65.7	75.5	78.7
	S2	10.1	63.2	64.6	62.6
	acc.	47.9	64.5	70.0	$(\lambda = 0.1)$ 70.7
Total average accuracy		58.0	63.4	67.4	69.5

Table 2: *Discrimination results in %.*

spectively), except for K-means (59.5%), than the ones achieved in [3] using Shütze's global vector-based approach (accuracy of 62.1% and 55.8% for K-means and the diagonal FGM respectively). Finally, the clusters obtained with the Gaussian model are more balanced than with K-means, for all the pseudowords.

6 Conclusion

A local vector-based model related to Schütze's approach is proposed for unsupervised sense discrimination. The main idea is to reduce the dimension of the high dimensional word vectors by means of linguistic facts. By a drastic reduction of the dimensionality (i.e. by a factor 100 as compared to Schütze's global model), the simplified assumptions on the Gaussian mixtures are not required anymore. Experimental results show that Gaussian mixtures with full covariance matrices does improve the results. The best performances are obtained when using the regularized

Mahalanobis distance. The Gaussian models lead to balanced clusters.

The approach proposed here is also linguistically coherent (use of collocation, small context window, taking into account prepositions) and its results can be explained by linguistic intuition (e.g. for the *rely-illustration* pseudoword for which the components have different parts of speech).

Several additional extensions will be considered in the future. We plan to add syntactic structure to the model: dimensions of the vector could be positional part-of-speech tags. The number of considered senses has to be studied: currently only binary sense discrimination is considered. We also plan to test the model on real ambiguous words.

References

[1] C. Archambeau, J. Lee, and M. Verleysen. On convergence problems of the *EM* algorithm for finite gaussian mixtures. In *11th ESANN Symp.*, pages 99–106, Bruges (Belgium), 2003.

[2] C. Archambeau and M. Verleysen. Fully nonparametric probability density function estimation with finite gaussian mixture models. In *7th ICPAR Conf.*, pages 81–84, Calcutta (India), 2003.

[3] M.-C. de Marneffe and P. Dupont. Comparative study of statistical word sense discrimination. In *7th JADT Conf.*, Louvain-la-Neuve (Belgium), 2004.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the *EM* algorithm. *J Roy. Stat. Soc. (B)*, 39:1–38, 1977.

[5] R. Duda, P. Hart, and D. Stork. *Pattern Classification.* Wiley, New York, 2001.

[6] T. Gaustad. Statistical corpus-based word sense disambiguation: Pseudowords vs real ambiguous words. *39th ACL/EACL – Student Research Workshop*, 2001.

[7] G. McLachlan and D. Peel. *Finite Mixture Models.* Wiley, 2000.

[8] A. Meillet. *Linguistique historique et linguistique générale*, volume 1. Champion, Paris, 1926.

[9] H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *1st NMLP Conf.*, 1994.

[10] H. Schütze. Automatic word sense discrimination. *Computational Linguistics*, 24:97–124, 1998.

[11] J. Sinclair. *Corpus, Concordance, Collocation.* Oxford University Press, 1991.

[12] J. Véronis. Sense tagging : does it make sense? In *Corpus Linguistics' 2001 Conference*, 2001.