A Kernel PCA Method for Superior Word Sense Disambiguation

Dekai WU[1] **Weifeng SU** **Marine CARPUAT**
dekai@cs.ust.hk weifeng@cs.ust.hk marine@cs.ust.hk

Human Language Technology Center
HKUST
Department of Computer Science
University of Science and Technology
Clear Water Bay, Hong Kong

Abstract

We introduce a new method for disambiguating word senses that exploits a nonlinear *Kernel Principal Component Analysis* (*KPCA*) technique to achieve accuracy superior to the best published individual models. We present empirical results demonstrating significantly better accuracy compared to the state-of-the-art achieved by either naïve Bayes or maximum entropy models, on Senseval-2 data. We also contrast against another type of kernel method, the support vector machine (SVM) model, and show that our KPCA-based model outperforms the SVM-based model. It is hoped that these highly encouraging first results on KPCA for natural language processing tasks will inspire further development of these directions.

1 Introduction

Achieving higher precision in supervised word sense disambiguation (WSD) tasks without resorting to *ad hoc* voting or similar ensemble techniques has become somewhat daunting in recent years, given the challenging benchmarks set by naïve Bayes models (e.g., Mooney (1996), Chodorow *et al.* (1999), Pedersen (2001), Yarowsky and Florian (2002)) as well as maximum entropy models (e.g., Dang and Palmer (2002), Klein and Manning (2002)). A good foundation for comparative studies has been established by the Senseval data and evaluations; of particular relevance here are the lexical sample tasks from Senseval-1 (Kilgarriff and Rosenzweig, 1999) and Senseval-2 (Kilgarriff, 2001).

We therefore chose this problem to introduce an efficient and accurate new word sense disambiguation approach that exploits a nonlinear Kernel PCA technique to make predictions implicitly based on generalizations over feature combinations. The technique is applicable whenever vector representations of a disambiguation task can be generated; thus many properties of our technique can be expected to be highly attractive from the standpoint of natural language processing in general.

In the following sections, we first analyze the potential of nonlinear principal components with respect to the task of disambiguating word senses. Based on this, we describe a full model for WSD built on KPCA. We then discuss experimental results confirming that this model outperforms state-of-the-art published models for Senseval-related lexical sample tasks as represented by (1) naïve Bayes models, as well as (2) maximum entropy models. We then consider whether other kernel methods—in particular, the popular SVM model—are equally competitive, and discover experimentally that KPCA achieves higher accuracy than the SVM model.

2 Nonlinear principal components and WSD

The *Kernel Principal Component Analysis* technique, or *KPCA*, is a nonlinear kernel method for extraction of nonlinear principal components from vector sets in which, conceptually, the n-dimensional input vectors are nonlinearly mapped from their original space R^n to a high-dimensional feature space F where linear PCA is performed, yielding a transform by which the input vectors can be mapped nonlinearly to a new set of vectors (Schölkopf *et al.*, 1998).

A major advantage of KPCA is that, unlike other common analysis techniques, as with other kernel methods it inherently takes *combinations* of predictive features into account when optimizing dimensionality reduction. For natural language problems in general, of course, it is widely recognized that significant accuracy gains can often be achieved by generalizing over relevant feature combinations (e.g., Kudo and Matsumoto (2003)). Another advantage of KPCA for the WSD task is that the dimensionality of the input data is generally very

[1] The author would like to thank the Hong Kong Research Grants Council (RGC) for supporting this research in part through grants RGC6083/99E, RGC6256/00E, and DAG03/04.EG09.

Table 1: Two of the Senseval-2 sense classes for the target word "art", from WordNet 1.7 (Fellbaum 1998).

Class	**Sense**
1	the creation of beautiful or significant things
2	a superior skill

large, a condition where kernel methods excel.

Nonlinear principal components (Diamantaras and Kung, 1996) may be defined as follows. Suppose we are given a training set of M pairs (x_t, c_t) where the observed vectors $x_t \in R^n$ in an n-dimensional input space X represent the context of the target word being disambiguated, and the correct class c_t represents the sense of the word, for $t = 1, .., M$. Suppose Φ is a nonlinear mapping from the input space R^n to the feature space F. Without loss of generality we assume the M vectors are centered vectors in the feature space, i.e., $\sum_{t=1}^{M} \Phi(x_t) = 0$; uncentered vectors can easily be converted to centered vectors (Schölkopf *et al.*, 1998). We wish to diagonalize the covariance matrix in F:

$$C = \frac{1}{M} \sum_{j=1}^{M} \Phi(x_j) \Phi^T(x_j) \tag{1}$$

To do this requires solving the equation $\lambda v = Cv$ for eigenvalues $\lambda \geq 0$ and eigenvectors $v \in F$. Because

$$Cv = \frac{1}{M} \sum_{j=1}^{M} (\Phi(x_j) \cdot v) \Phi(x_j) \tag{2}$$

we can derive the following two useful results. First,

$$\lambda (\Phi(x_t) \cdot v) = \Phi(x_t) \cdot Cv \tag{3}$$

for $t = 1, .., M$. Second, there exist α_i for $i = 1, ..., M$ such that

$$v = \sum_{i=1}^{M} \alpha_i \Phi(x_i) \tag{4}$$

Combining (1), (3), and (4), we obtain

$$\begin{aligned} & M\lambda \sum_{i=1}^{M} \alpha_i (\Phi(x_t) \cdot \Phi(x_i)) \\ &= \sum_{i=1}^{M} \alpha_i (\Phi(x_t) \cdot \sum_{j=1}^{M} \Phi(x_j)) (\Phi(x_j) \cdot \Phi(x_i)) \end{aligned}$$

for $t = 1, .., M$. Let $\hat{K}$ be the $M \times M$ matrix such that

$$\hat{K}_{ij} = \Phi(x_i) \cdot \Phi(x_j) \tag{5}$$

and let $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \ldots \geq \hat{\lambda}_M$ denote the eigenvalues of $\hat{K}$ and $\hat{\alpha}^1$,..., $\hat{\alpha}^M$ denote the corresponding complete set of normalized eigenvectors, such that $\hat{\lambda}_t(\hat{\alpha}^t \cdot \hat{\alpha}^t) = 1$ when $\hat{\lambda}_t > 0$. Then the lth nonlinear principal component of any test vector x_t is defined as

$$y_t^l = \sum_{i=1}^{M} \hat{\alpha}_i^l (\Phi(x_i) \cdot \Phi(x_t)) \tag{6}$$

where $\hat{\alpha}_i^l$ is the lth element of $\hat{\alpha}^l$.

To illustrate the potential of nonlinear principal components for WSD, consider a simplified disambiguation example for the ambiguous target word "art", with the two senses shown in Table 1. Assume a training corpus of the eight sentences as shown in Table 2, adapted from Senseval-2 English lexical sample corpus. For each sentence, we show the feature set associated with that occurrence of "art" and the correct sense class. These eight occurrences of "art" can be transformed to a binary vector representation containing one dimension for each feature, as shown in Table 3.

Extracting nonlinear principal components for the vectors in this simple corpus results in nonlinear generalization, reflecting an implicit consideration of combinations of features. Table 3 shows the first three dimensions of the principal component vectors obtained by transforming each of the eight training vectors x_t into (a) principal component vectors z_t using the linear transform obtained via PCA, and (b) nonlinear principal component vectors y_t using the nonlinear transform obtained via KPCA as described below.

Similarly, for the test vector x_9, Table 4 shows the first three dimensions of the principal component vectors obtained by transforming it into (a) a principal component vector z_9 using the linear PCA transform obtained from training, and (b) a nonlinear principal component vector y_9 using the nonlinear KPCA transform obtained obtained from training. The vector similarities in the KPCA-transformed space can be quite different from those in the PCA-transformed space. This causes the KPCA-based model to be able to make the correct class prediction, whereas the PCA-based model makes the

Table 2: A tiny corpus for the target word "art", adapted from the Senseval-2 English lexical sample corpus (Kilgarriff 2001), together with a tiny example set of features. The training and testing examples can be represented as a set of binary vectors: each row shows the correct class c for an observed vector x of five dimensions.

	TRAINING	design/N	media/N	the/DT	entertainment/N	world/N	**Class**
x_1	He studies **art** in London.						1
x_2	Punch's weekly guide to the world of the **arts**, entertainment, media and more.		1	1	1	1	
x_3	All such studies have influenced every form of **art**, design, and entertainment in some way.	1			1		1
x_4	Among the technical **arts** cultivated in some continental schools that began to affect England soon after the Norman Conquest were those of measurement and calculation.			1			2
x_5	The **Art** of Love.			1			2
x_6	Indeed, the **art** of doctoring does contribute to better health results and discourages unwarranted malpractice litigation.			1			2
x_7	Countless books and classes teach the **art** of asserting oneself.			1			2
x_8	Pop **art** is an example.						1
	TESTING						
x_9	In the world of design **arts** particularly, this led to appointments made for political rather than academic reasons.	1		1		1	1

wrong class prediction.

What permits KPCA to apply stronger generalization biases is its implicit consideration of *combinations* of feature information in the data distribution from the high-dimensional training vectors. In this simplified illustrative example, there are just five input dimensions; the effect is stronger in more realistic high dimensional vector spaces. Since the KPCA transform is computed from unsupervised training vector data, and extracts generalizations that are subsequently utilized during supervised classification, it is quite possible to combine large amounts of unsupervised data with reasonable smaller amounts of supervised data.

It can be instructive to attempt to interpret this example graphically, as follows, even though the interpretation in three dimensions is severely limiting. Figure 1(a) depicts the eight original observed training vectors x_t in the first three of the five dimensions; note that among these eight vectors, there happen to be only four unique points when restricting our view to these three dimensions. Ordinary linear PCA can be straightforwardly seen as projecting the original points onto the principal axis,

Table 3: The original observed training vectors (showing only the first three dimensions) and their first three principal components as transformed via PCA and KPCA.

	Observed vectors	**PCA-transformed vectors**	**KPCA-transformed vectors**	**Class**
t	(x_t^1, x_t^2, x_t^3)	(z_t^1, z_t^2, z_t^3)	(y_t^1, y_t^2, y_t^3)	c_t
1	(0, 0, 0)	(-1.961, 0.2829, 0.2014)	(0.2801, -1.005, -0.06861)	1
2	(0, 1, 1)	(1.675, -1.132, 0.1049)	(1.149, 0.02934, 0.322)	1
3	(1, 0, 0)	(-0.367, 1.697, -0.2391)	(0.8209, 0.7722, -0.2015)	1
4	(0, 0, 1)	(-1.675, -1.132, -0.1049)	(-1.774, -0.1216, 0.03258)	2
5	(0, 0, 1)	(-1.675, -1.132, -0.1049)	(-1.774, -0.1216, 0.03258)	2
6	(0, 0, 1)	(-1.675, -1.132, -0.1049)	(-1.774, -0.1216, 0.03258)	2
7	(0, 0, 1)	(-1.675, -1.132, -0.1049)	(-1.774, -0.1216, 0.03258)	2
8	(0, 0, 0)	(-1.961, 0.2829, 0.2014)	(0.2801, -1.005, -0.06861)	1

Table 4: Testing vector (showing only the first three dimensions) and its first three principal components as transformed via the trained PCA and KPCA parameters. The PCA-based and KPCA-based sense class predictions disagree.

	Observed vectors	**PCA-transformed vectors**	**KPCA-transformed vectors**	**Predicted Class**	**Correct Class**
t	(x_t^1, x_t^2, x_t^3)	(z_t^1, z_t^2, z_t^3)	(y_t^1, y_t^2, y_t^3)	$\hat{c}_t$	c_t
9	(1, 0, 1)	(-0.3671, -0.5658, -0.2392)		2	1
9	(1, 0, 1)		(4e-06, 8e-07, 1.111e-18)	1	1

as can be seen for the case of the first principal axis in Figure 1(b). Note that in this space, the sense 2 instances are surrounded by sense 1 instances. We can traverse each of the projections onto the principal axis in linear order, simply by visiting each of the first principal components z_t^1 along the principle axis in order of their values, i.e., such that

$$z_1^1 \leq z_8^1 \leq z_4^1 \leq z_5^1 \leq z_6^1 \leq z_7^1 \leq z_2^1 \leq z_3^1 \leq z_9^1$$

It is significantly more difficult to visualize the nonlinear principal components case, however. Note that in general, there may not exist *any* principal axis in X, since an inverse mapping from F may not exist. If we attempt to follow the same procedure to traverse each of the projections onto the first principal axis as in the case of linear PCA, by considering each of the first principal components y_t^1 in order of their value, i.e., such that

$$y_4^1 \leq y_5^1 \leq y_6^1 \leq y_7^1 \leq y_9^1 \leq y_1^1 \leq y_8^1 \leq y_3^1 \leq y_2^1$$

then we must arbitrarily select a "quasi-projection" direction for each y_t^1 since there is no actual principal axis toward which to project. This results in a "quasi-axis" roughly as shown in Figure 1(c) which, though not precisely accurate, provides some idea as to how the nonlinear generalization capability allows the data points to be grouped by principal components reflecting nonlinear patterns in the data distribution, in ways that linear PCA cannot do. Note that in this space, the sense 1 instances are already better separated from sense 2 data points. Moreover, unlike linear PCA, there may be up to M of the "quasi-axes", which may number far more than five. Such effects can become pronounced in the high dimensional spaces are actually used for real word sense disambiguation tasks.

3 A KPCA-based WSD model

To extract nonlinear principal components efficiently, note that in both Equations (5) and (6) the explicit form of $\Phi\left(x_i\right)$ is required only in the form of $(\Phi\left(x_i\right) \cdot \Phi\left(x_j\right))$, i.e., the dot product of vectors in F. This means that we can calculate the nonlinear principal components by substituting a kernel function $k(x_i, x_j)$ for $(\Phi(\,x_i) \cdot \Phi(x_j\,))$ in Equations (5) and (6) without knowing the mapping Φ explicitly; instead, the mapping Φ is implicitly defined by the kernel function. It is always possible to construct a mapping into a space where k acts as a dot product so long as k is a continuous kernel of a positive integral operator (Schölkopf *et al.*, 1998).

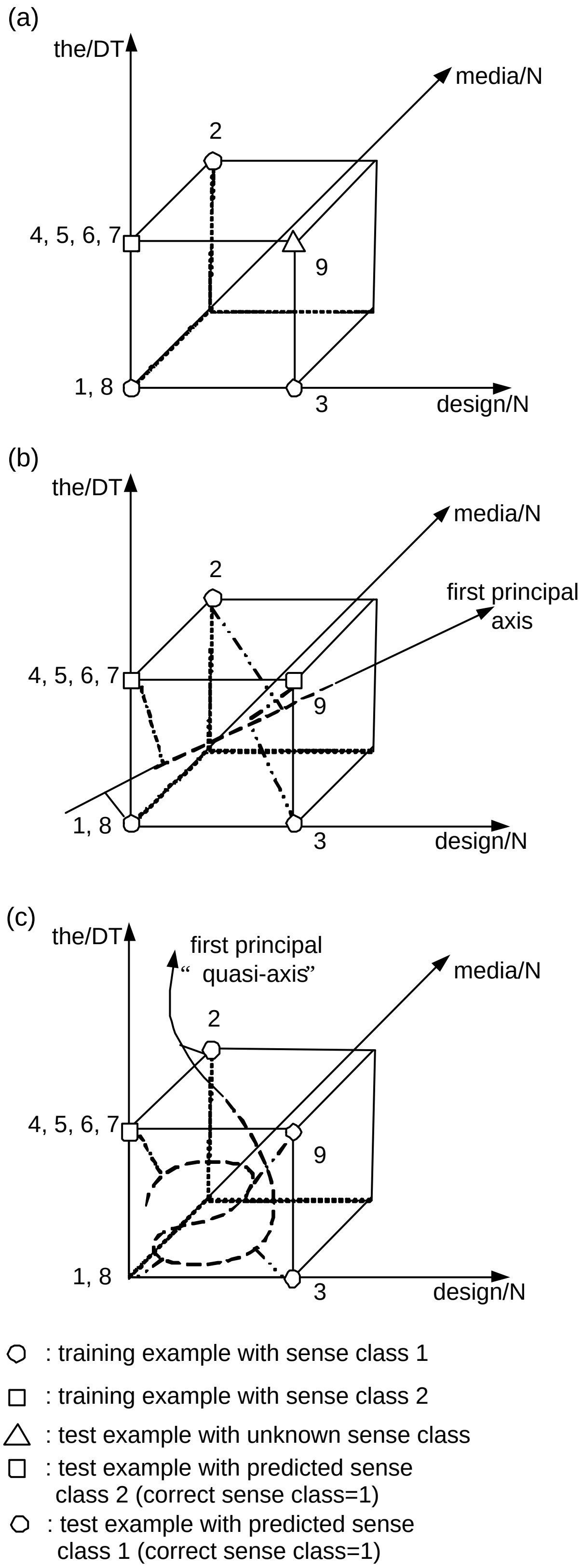
(a)
the/DT
media/N
2
4, 5, 6, 7
9
1, 8
3
design/N
(b)
the/DT
media/N
2
first principal axis
4, 5, 6, 7
9
1, 8
3
design/N
(c)
the/DT
first principal “ quasi-axis”
media/N
2
4, 5, 6, 7
9
1, 8
3
design/N
○ : training example with sense class 1
□ : training example with sense class 2
△ : test example with unknown sense class
□ : test example with predicted sense class 2 (correct sense class=1)
○ : test example with predicted sense class 1 (correct sense class=1)

Figure 1: Original vectors, PCA projections, and KPCA “quasi-projections” (see text).

Table 5: Experimental results showing that the KPCA-based model performs significantly better than naïve Bayes and maximum entropy models. Significance intervals are computed via bootstrap resampling.

WSD Model	**Accuracy**	**Sig. Int.**
naïve Bayes	63.3%	+/-0.91%
maximum entropy	63.8%	+/-0.79%
KPCA-based model	**65.8%**	+/-0.79%

Thus we train the KPCA model using the following algorithm:

1. Compute an $M \times M$ matrix $\hat{K}$ such that
$$\hat{K}_{ij} = k(x_i, x_j) \tag{7}$$
2. Compute the eigenvalues and eigenvectors of matrix $\hat{K}$ and normalize the eigenvectors. Let $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \ldots \geq \hat{\lambda}_M$ denote the eigenvalues and $\hat{\alpha}^1$,..., $\hat{\alpha}^M$ denote the corresponding complete set of normalized eigenvectors.

To obtain the sense predictions for test instances, we need only transform the corresponding vectors using the trained KPCA model and classify the resultant vectors using nearest neighbors. For a given test instance vector x, its lth nonlinear principal component is

$$y_t^l = \sum_{i=1}^{M} \hat{\alpha}_i^l k(x_i, x_t) \tag{8}$$

where $\hat{\alpha}_i^l$ is the ith element of $\hat{\alpha}^l$.

For our disambiguation experiments we employ a polynomial kernel function of the form $k(x_i, x_j) = (x_i \cdot x_j)^d$, although other kernel functions such as gaussians could be used as well. Note that the degenerate case of $d = 1$ yields the dot product kernel $k(x_i, x_j) = (x_i{\cdot}x_j)$ which covers linear PCA as a special case, which may explain why KPCA always outperforms PCA.

4 Experiments

4.1 KPCA versus naïve Bayes and maximum entropy models

We established two baseline models to represent the state-of-the-art for individual WSD models: (1) naïve Bayes, and (2) maximum entropy models. The naïve Bayes model was found to be the most accurate classifier in a comparative study using a subset of Senseval-2 English lexical sample data by Yarowsky and Florian (2002). However, the maximum entropy (Jaynes, 1978) was found to yield higher accuracy than naïve Bayes in a subsequent comparison by Klein and Manning (2002), who used a different subset of either Senseval-1 or Senseval-2 English lexical sample data. To control for data variation, we built and tuned models of both kinds. Note that our objective in these experiments is to understand the performance and characteristics of KPCA relative to other *individual* methods. It is not our objective here to compare against voting or other ensemble methods which, though known to be useful in practice (e.g., Yarowsky *et al.* (2001)), would not add to our understanding.

To compare as evenly as possible, we employed features approximating those of the "feature-enhanced naïve Bayes model" of Yarowsky and Florian (2002), which included position-sensitive, syntactic, and local collocational features. The models in the comparative study by Klein and Manning (2002) did not include such features, and so, again for consistency of comparison, we experimentally verified that our maximum entropy model (a) consistently yielded higher scores than when the features were not used, and (b) consistently yielded higher scores than naïve Bayes using the same features, in agreement with Klein and Manning (2002). We also verified the maximum entropy results against several different implementations, using various smoothing criteria, to ensure that the comparison was even.

Evaluation was done on the Senseval 2 English lexical sample task. It includes 73 target words, among which nouns, adjectives, adverbs and verbs. For each word, training and test instances tagged with WordNet senses are provided. There are an average of 7.8 senses per target word type. On average 109 training instances per target word are available. Note that we used the set of sense classes from Senseval's ”fine-grained” rather than ”coarse-grained” classification task.

The KPCA-based model achieves the highest accuracy, as shown in Table 5, followed by the maximum entropy model, with naïve Bayes doing the poorest. Bear in mind that *all* of these models are significantly more accurate than any of the other reported models on Senseval. "Accuracy" here refers to both precision and recall since disambiguation of all target words in the test set is attempted. Results are statistically significant at the 0.10 level, using bootstrap resampling (Efron and Tibshirani, 1993); moreover, we consistently witnessed the same level of accuracy gains from the KPCA-based model over many variations of the experiments.

Table 6: Experimental results comparing the KPCA-based model versus the SVM model.

WSD Model	**Accuracy**	**Sig. Int.**
SVM-based model	65.2%	+/-1.00%
KPCA-based model	**65.8%**	+/-0.79%

4.2 KPCA versus SVM models

Support vector machines (e.g., Vapnik (1995), Joachims (1998)) are a different kind of kernel method that, unlike KPCA methods, have already gained high popularity for NLP applications (e.g., Takamura and Matsumoto (2001), Isozaki and Kazawa (2002), Mayfield *et al.* (2003)) including the word sense disambiguation task (e.g., Cabezas *et al.* (2001)). Given that SVM and KPCA are both kernel methods, we are frequently asked whether SVM-based WSD could achieve similar results.

To explore this question, we trained and tuned an SVM model, providing the same rich set of features and also varying the feature representations to optimize for SVM biases. As shown in Table 6, the highest-achieving SVM model is also able to obtain higher accuracies than the naïve Bayes and maximum entropy models. However, in all our experiments the KPCA-based model consistently outperforms the SVM model (though the margin falls within the statistical significance interval as computed by bootstrap resampling for this single experiment). The difference in KPCA and SVM performance is not surprising given that, aside from the use of kernels, the two models share little structural resemblance.

4.3 Running times

Training and testing times for the various model implementations are given in Table 7, as reported by the Unix `time` command. Implementations of all models are in C++, but the level of optimization is not controlled. For example, no attempt was made to reduce the training time for naïve Bayes, or to reduce the testing time for the KPCA-based model. Nevertheless, we can note that in the operating range of the Senseval lexical sample task, the running times of the KPCA-based model are roughly within the same order of magnitude as for naïve Bayes or maximum entropy. On the other hand, training is much faster than the alternative kernel method based on SVMs. However, the KPCA-based model's times could be expected to suffer in situations where significantly larger amounts of

Table 7: Comparison of training and testing times for the different WSD model implementations.

WSD Model	Training time [CPU sec]	Testing time [CPU sec]
naïve Bayes	103.41	16.84
maximum entropy	104.62	59.02
SVM-based model	5024.34	16.21
KPCA-based model	216.50	128.51

training data are available.

5 Conclusion

This work represents, to the best of our knowledge, the first application of Kernel PCA to a true natural language processing task. We have shown that a KPCA-based model can significantly outperform state-of-the-art results from both naïve Bayes as well as maximum entropy models, for supervised word sense disambiguation. The fact that our KPCA-based model outperforms the SVM-based model indicates that kernel methods other than SVMs deserve more attention. Given the theoretical advantages of KPCA, it is our hope that this work will encourage broader recognition, and further exploration, of the potential of KPCA modeling within NLP research.

Given the positive results, we plan next to combine large amounts of unsupervised data with reasonable smaller amounts of supervised data such as the Senseval lexical sample. Earlier we mentioned that one of the promising advantages of KPCA is that it computes the transform purely from unsupervised training vector data. We can thus make use of the vast amounts of cheap unannotated data to augment the model presented in this paper.

References

Clara Cabezas, Philip Resnik, and Jessica Stevens. Supervised sense tagging using support vector machines. In *Proceedings of Senseval-2, Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 59–62, Toulouse, France, July 2001. SIGLEX, Association for Computational Linguistics.

Martin Chodorow, Claudia Leacock, and George A. Miller. A topical/local classifier for word sense identification. *Computers and the Humanities*, 34(1-2):115–120, 1999. Special issue on SENSEVAL.

Hoa Trang Dang and Martha Palmer. Combining contextual features for word sense disambiguation. In *Proceedings of the SIGLEX/SENSEVAL Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, pages 88–94, Philadelphia, July 2002. SIGLEX, Association for Computational Linguistics.

Konstantinos I. Diamantaras and Sun Yuan Kung. *Principal Component Neural Networks*. Wiley, New York, 1996.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, 1993.

Hideki Isozaki and Hideto Kazawa. Efficient support vector classifiers for named entity recognition. In *Proceedings of COLING-2002*, pages 390–396, Taipei, 2002.

E.T. Jaynes. *Where do we Stand on Maximum Entropy?* MIT Press, Cambridge MA, 1978.

Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142, 1998.

Adam Kilgarriff and Joseph Rosenzweig. Framework and results for English Senseval. *Computers and the Humanities*, 34(1):15–48, 1999. Special issue on SENSEVAL.

Adam Kilgarriff. English lexical sample task description. In *Proceedings of Senseval-2, Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 17–20, Toulouse, France, July 2001. SIGLEX, Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. Conditional structure versus conditional estimation in NLP models. In *Proceedings of EMNLP-2002, Conference on Empirical Methods in Natural Language Processing*, pages 9–16, Philadelphia, July 2002. SIGDAT, Association for Computational Linguistics.

Taku Kudo and Yuji Matsumoto. Fast methods for kernel-based text analysis. In *Proceedings of* *the 41set Annual Meeting of the Asoociation for Computational Linguistics*, pages 24–31, 2003.

James Mayfield, Paul McNamee, and Christine Piatko. Named entity recognition using hundreds of thousands of features. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 184–187, Edmonton, Canada, 2003.

Raymond J. Mooney. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Philadelphia, May 1996. SIGDAT, Association for Computational Linguistics.

Ted Pedersen. Machine learning with lexical features: The Duluth approach to SENSEVAL-2. In *Proceedings of Senseval-2, Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 139–142, Toulouse, France, July 2001. SIGLEX, Association for Computational Linguistics.

Bernhard Schölkopf, Alexander Smola, and Klaus-Rober Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5), 1998.

Hiroya Takamura and Yuji Matsumoto. Feature space restructuring for SVMs with application to text categorization. In *Proceedings of EMNLP-2001, Conference on Empirical Methods in Natural Language Processing*, pages 51–57, 2001.

Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

David Yarowsky and Radu Florian. Evaluating sense disambiguation across diverse parameter spaces. *Natural Language Engineering*, 8(4):293–310, 2002.

David Yarowsky, Silviu Cucerzan, Radu Florian, Charles Schafer, and Richard Wicentowski. The Johns Hopkins SENSEVAL2 system descriptions. In *Proceedings of Senseval-2, Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 163–166, Toulouse, France, July 2001. SIGLEX, Association for Computational Linguistics.