Augmenting Ensemble Classification for Word Sense Disambiguation with a Kernel PCA Model

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

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

Abstract

The HKUST word sense disambiguation systems benefit from a new nonlinear Kernel Principal Component Analysis (KPCA) based disambiguation technique. We discuss and analyze results from the Senseval-3 English, Chinese, and Multilingual Lexical Sample data sets. Among an ensemble of four different kinds of voted models, the KPCA-based model, along with the maximum entropy model, outperforms the boosting model and naïve Bayes model. Interestingly, while the KPCA-based model typically achieves close or better accuracy than the maximum entropy model, nevertheless a comparison of predicted classifications shows that it has a significantly different bias. This characteristic makes it an excellent voter, as confirmed by results showing that removing the KPCA-based model from the ensemble generally degrades performance.

1 Introduction

Classifier combination has become a standard architecture for shared task evaluations in word sense disambiguation (WSD), named entity recognition, and similar problems that can naturally be cast as classification problems. Voting is the most common method of combination, having proven to be remarkably effective yet simple.

A key problem in improving the accuracy of such ensemble classification systems is to find new voting models that (1) exhibit significantly different prediction biases from the models already voting, and yet (2) attain stand-alone classification accuracies that are as good or better. When either of these conditions is not met, adding the new voting model typically degrades the accuracy of the ensemble instead of helping it.

In this work, we investigate the potential of one promising new disambiguation model with respect to augmenting our existing ensemble combining a maximum entropy model, a boosting model, and a naïve Bayes model—a combination representing some of the best stand-alone WSD models currently known. The new WSD model, proposed by Wu *et al.* (2004), is a method for disambiguating word senses that exploits a nonlinear *Kernel Principal Component Analysis* (*KPCA*) technique. That the KPCA-based model could potentially be a good candidate for a new voting model is suggested by Wu *et al.*'s empirical results showing that it yielded higher accuracies on Senseval-2 data sets than other models that included maximum entropy, naïve Bayes, and SVM based models.

In the following sections, we begin with a description of the experimental setup, which utilizes a number of individual classifiers in a voting ensemble. We then describe the KPCA-based model to be added to the baseline ensemble. The accuracy results of the three submitted models are examined, and also the individual voting models are compared. Subsequently, we analyze the degree of difference in voting bias of the KPCA-based model from the others, and finally show that this does indeed usually lead to accuracy gains in the voting ensemble.

2 Experimental setup

2.1 Tasks evaluated

We performed experiments on the following lexical sample tasks from Senseval-3:

English (fine). The English lexical sample task includes 57 target words (32 verbs, 20 nouns and 5 adjectives). For each word, training and test instances tagged with WordNet senses are provided. There are an average of 8.5 senses per target word type, ranging from 3 to 23. On average, 138 training instances per target word are available.

English (coarse). This modified evaluation of the preceding task employs a sense map that groups fine-grained sense distinctions into the same coarse-grained sense.

Chinese. The Chinese lexical sample task includes 21 target words. For each word, several

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

senses are defined using the HowNet knowledge base. There are an average of 3.95 senses per target word type, ranging from 2 to 8. Only about 37 training instances per target word are available.

Multilingual (t). The Multilingual (t) task is defined similarly to the English lexical sample task, except that the word senses are the translations into Hindi, rather than WordNet senses. The Multilingual (t) task requires finding the Hindi sense for 31 English target word types. There are an average of 7.54 senses per target word type, ranging from 3 to 16. A relatively large training set is provided (more than 260 training instances per word on average).

Multilingual (ts). The Multilingual (ts) task uses a different data set of 10 target words and provides the correct English sense of the target word for both training and testing. There are an average of 6.2 senses per target word type, ranging from 3 to 11. The training set for this subtask was smaller, with about 150 training instances per target word.

2.2 Ensemble classification

The WSD models presented here consist of ensembles utilizing various combinations of four voting models, as follows. Some of these component models were also evaluated on other Senseval-3 tasks: the Basque, Catalan, Italian, and Romanian Lexical Sample tasks (Wicentowski *et al.*, 2004), as well as Semantic Role Labeling (Ngai *et al.*, 2004).

The first voting model, a naïve Bayes model, was built as Yarowsky and Florian (2002) found this model to be the most accurate classifier in a comparative study on a subset of Senseval-2 English lexical sample data.

The second voting model, a maximum entropy model (Jaynes, 1978), was built as Klein and Manning (2002) found that it yielded higher accuracy than naïve Bayes in a subsequent comparison of WSD performance. However, note that a different subset of either Senseval-1 or Senseval-2 English lexical sample data was used.

The third voting model, a boosting model (Freund and Schapire, 1997), was built as boosting has consistently turned in very competitive scores on related tasks such as named entity classification (Carreras *et al.*, 2002)(Wu *et al.*, 2002). Specifically, we employed an AdaBoost.MH model (Schapire and Singer, 2000), which is a multi-class generalization of the original boosting algorithm, with boosting on top of decision stump classifiers (decision trees of depth one).

The fourth voting model, the KPCA-based model, is described below.

All classifier models were selected for their ability to able to handle large numbers of sparse features, many of which may be irrelevant. Moreover, the maximum entropy and boosting models are known to be well suited to handling features that are highly interdependent.

2.3 Controlled feature set

In order to facilitate a controlled comparison across the individual voting models, the same feature set was employed for all classifiers. The features are as described by Yarowsky and Florian (2002) in their "feature-enhanced naïve Bayes model", with position-sensitive, syntactic, and local collocational features.

2.4 The KPCA-based WSD model

We briefly summarize the KPCA-based model here; for full details including illustrative examples and graphical interpretation, please refer to Wu *et al.* (2004).

Kernel PCA Kernel Principal Component Analysis is a nonlinear kernel method for extracting nonlinear principal components from vector sets where, 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).

As with other kernel methods, a major advantage of KPCA over other common analysis techniques is that it can inherently take *combinations* of predictive features into account when optimizing dimensionality reduction. For WSD and indeed many natural language tasks, significant accuracy gains can often be achieved by generalizing over relevant feature combinations (see, e.g., Kudo and Matsumoto (2003)). A further advantage of KPCA in the context of the WSD problem is that the dimensionality of the input data is generally very large, a condition where kernel methods excel.

Nonlinear principal components (Diamantaras and Kung, 1996) are 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:

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

To do this requires solving the equation $\lambda v = Cv$ for eigenvalues $\lambda \geq 0$ and eigenvectors $v \in R^n \backslash \{0\}$. Because

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

we can derive the following two useful results. First,

$$\lambda\left(\Phi(x_t)\cdot v\right) = \Phi\left(x_t\right)\cdot Cv \quad (3)$$

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

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

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

$$M\lambda\sum_{i=1}^{M}\alpha_i\left(\Phi(x_t)\cdot\Phi(x_i)\right)$$
$$=\sum_{i=1}^{M}\alpha_i(\Phi\left(x_t\right)\cdot\sum_{j=1}^{M}\Phi\left(x_j\right))\left(\Phi(x_j)\cdot\Phi(x_i)\right)$$

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

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

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 *l*th nonlinear principal component of any test vector x_t is defined as

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

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

See Wu *et al.* (2004) for a possible geometric interpretation of the power of the nonlinearity.

WSD using KPCA In order to extract nonlinear principal components efficiently, first 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).

Thus we train the KPCA model using the following algorithm:

1. Compute an $M \times M$ matrix $\hat{K}$ such that
$$\hat{K}_{ij} = k(x_i, x_j) \quad (7)$$
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 *l*th nonlinear principal component is

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

where $\hat{\alpha}_i^l$ is the *i*th 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.

3 Results and discussion

3.1 Accuracy

Table 1 summarizes the results of the submitted systems along with the individual voting models. Since our models attempted to disambiguate all test instances, we report accuracy (precision and recall being equal). Earlier experiments on Senseval-2 data showed that the KPCA-based model significantly outperformed both naïve Bayes and maximum entropy models (Wu *et al.*, 2004). On the Senseval-3 data, the maximum entropy model fares slightly better: it remains significantly worse on the Multilingual (ts) task, but achieves statistically the same accuracy on the English (fine) task and is slightly

Table 1: Comparison of accuracy results for various HKUST ensemble and individual models on Senseval-3 Lexical Sample tasks, confirming the high accuracy of the KPCA-based model. All test instances were attempted. (Bold model names were the systems entered.)

	English (fine)	**English (coarse)**	**Chinese**	**Multilingual (t)**	**Multilingual (ts)**
HKUST_comb2 (me, boost, nb, kpca)	71.4	78.6	66.2	62.0	63.8
HKUST_comb (me, boost, kpca)	70.9	78.1	66.5	61.4	63.8
HKUST_me	69.3	76.4	64.4	60.6	60.8
HKUST_kpca	69.2	-	63.6	60.0	63.3
HKUST_boost	67.0	-	64.1	57.3	60.3
HKUST_nb	64.3	-	60.4	57.3	56.8

Table 2: Confusion matrices showing that the KPCA-based model votes very differently from the other models on the Senseval-3 Lexical Sample tasks. Percentages representing disagreement between KPCA and other voting models are shown in bold.

	kpca vs:		**me**			**boost**			**nb**	
task			incorrect	correct		incorrect	correct		incorrect	correct
English	incorrect		24.14%	**6.62%**		21.60%	**9.15%**		21.04%	**9.71%**
(fine)	correct		**6.59%**	62.65%		**11.38%**	57.86%		**14.63%**	54.61%
Chinese	incorrect		24.01%	**12.40%**		22.96%	**13.46%**		26.65%	**9.76%**
	correct		**11.61%**	51.98%		**12.93%**	50.66%		**12.93%**	50.66%
Multilingual	incorrect		32.71%	**7.33%**		32.04%	**8.01%**		30.54%	**9.51%**
(t)	correct		**6.74%**	53.22%		**10.63%**	49.33%		**12.20%**	47.75%
Multilingual	incorrect		33.17%	**3.52%**		31.66%	**5.03%**		30.15%	**6.53%**
(ts)	correct		**6.03%**	57.29%		**8.04%**	55.28%		**13.07%**	50.25%

more accurate on the Multilingual (t) task. For unknown reasons—possibly the very small number of training instances per Chinese target word, as mentioned earlier—there is an exception on the Chinese task, where boosting outperforms the KPCA-based model. We are investigating the possible causes. The naïve Bayes model remains significantly worse under all conditions.

3.2 Differentiated voting bias

For a new voting model to raise the accuracy of an existing classifier ensemble, it is not only important that the new voting model achieve accuracy comparable to the other voters, as shown above, but also that it provides a significantly differentiated prediction bias than the other voters. Otherwise, the accuracy is typically hurt rather than helped by the new voting model.

To examine whether the KPCA-based model satisfies this requirement, we compared its predictions against each of the other classifiers (for those tasks where we have been given the answer key). Table 2 shows nine confusion matrices revealing the percentage of instances where the KPCA-based model votes differently from one of the other voters. The disagreement between KPCA and the other voting models ranges from 6.03% to 14.63%, as shown by the bold entries in the confusion matrices. Note that where there is disagreement, the KPCA-based model predicts the correct sense with significantly higher accuracy, in nearly all cases.

3.3 Voting effectiveness

The KPCA-based model exhibits the accuracy and differentiation characteristics requisite for an effective additional voter, as shown in the foregoing sec-

Table 3: Comparison of the accuracies for the voting ensembles with and without the KPCA voter, confirming that adding the KPCA-based model to the voting ensemble always helps on Senseval-3 Lexical Sample tasks.

	English (fine)	**English (coarse)**	**Chinese**	**Multilingual (t)**	**Multilingual (ts)**
HKUST_comb3 (me, boost, nb)	71.2	-	67.5	60.6	60.8
HKUST_comb2 (me, boost, nb, kpca)	71.4	78.6	66.2	62.0	63.8

tions. To verify that adding the KPCA-based model to the voting ensemble indeed improves accuracy, we compared our voting ensemble's accuracies to that obtained with KPCA removed. The results, shown in Table 3, confirm that the KPCA-based model generally helps on Senseval-3 Lexical Sample tasks. The only exception is on Chinese, due to the aforementioned anomaly of boosting outperforming KPCA on that task. In the Multilingual (t) and (ts) cases, the improvement in accuracy is significant.

4 Conclusion

We have described our word sense disambiguation system and its performance on the Senseval-3 English, Chinese, and Multilingual Lexical Sample tasks. The system consists of an ensemble classifier utilizing combinations of maximum entropy, boosting, naïve Bayes, and a new Kernel PCA based model.

We have demonstrated that our new model based on Kernel PCA is, along with maximum entropy, one of the most accurate stand-alone models voting in the ensemble, as evaluated under carefully controlled to ensure the same optimized feature set across all models being compared. Moreover, we have shown that the KPCA model exhibits a significantly different classification bias, a characteristic that makes it a valuable voter in an ensemble. The results confirm that accuracy is generally improved by the addition of the KPCA-based model.

References

Xavier Carreras, Lluí s Màrques, and Lluí s Padró. Named entity extraction using AdaBoost. In Dan Roth and Antal van den Bosch, editors, *Proceedings of CoNLL-2002*, pages 167–170, Taipei, Taiwan, 2002.

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

Yoram Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Journal of Computer and System Sciences, 55(1)*, pages 119–139, 1997.

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

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 41st Annual Meeting of the Association for Computational Linguistics*, pages 24–31, 2003.

Grace Ngai, Dekai Wu, Marine Carpuat, Chi-Shing Wang, and Chi-Yung Wang. Semantic role labeling with boosting, SVMs, maximum entropy, SNOW, and decision lists. In *Proceedings of Senseval-3, Third International Workshop on Evaluating Word Sense Disambiguation Systems*, Barcelona, July 2004. SIGLEX, Association for Computational Linguistics.

Robert E. Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. In *Machine Learning, 39(2/3)*, pages 135–168, 2000.

Bernhard Sch¨olkopf, Alexander Smola, and Klaus-Rober M¨uller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5), 1998.

Richard Wicentowski, Grace Ngai, Dekai Wu, Marine Carpuat, Emily Thomforde, and Adrian Packel. Joining forces to resolve lexical ambiguity: East meets West in Barcelona. In *Proceedings of Senseval-3, Third International Workshop on Evaluating Word Sense Disambiguation Systems*, Barcelona, July 2004. SIGLEX, Association for Computational Linguistics.

Dekai Wu, Grace Ngai, Marine Carpuat, Jeppe Larsen, and Yongsheng Yang. Boosting for named entity recognition. In Dan Roth and Antal van den Bosch, editors, *Proceedings of CoNLL-2002*, pages 195–198. Taipei, Taiwan, 2002.

Dekai Wu, Weifeng Su, and Marine Carpuat. A Kernel PCA method for superior word sense disambiguation. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, July 2004.

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