DOI: 10.1017/S1351324902002978 Printed in the United Kingdom

Combining classifiers for word sense disambiguation

RADU FLORIAN, SILVIU CUCERZAN, CHARLES SCHAFER and DAVID YAROWSKY

Department of Computer Science and Center for Language and Speech Processing
Johns Hopkins University, MD 21218, USA
e-mail: {rflorian,silviu,cschafer,yarowsky}@cs.jhu.edu

(*Received 19 November 2001; revised 20 June 2002*)

Abstract

Classifier combination is an effective and broadly useful method of improving system performance. This article investigates in depth a large number of both well-established and novel classifier combination approaches for the word sense disambiguation task, studied over a diverse classifier pool which includes feature-enhanced Naïve Bayes, Cosine, Decision List, Transformation-based Learning and MMVC classifiers. Each classifier has access to the same rich feature space, comprised of distance weighted bag-of-lemmas, local ngram context and specific syntactic relations, such as Verb-Object and Noun-Modifier. This study examines several key issues in system combination for the word sense disambiguation task, ranging from algorithmic structure to parameter estimation. Experiments using the standard SENSEVAL2 lexical-sample data sets in four languages (English, Spanish, Swedish and Basque) demonstrate that the combination system obtains a significantly lower error rate when compared with other systems participating in the SENSEVAL2 exercise, yielding state-of-the-art performance on these data sets.

1 Introduction

Classifier combination has been studied intensively in the last decade, and has been shown to be successful in improving performance on diverse applications (Brill and Wu 1998; van Halteren, Zavrel and Daelemans 1998; Kilgarriff and Rosenzweig 2000; Pedersen 2000; Stevenson and Wilks 2001).

The intuition behind classifier combination is that individual classifiers have different strengths and perform well on different subtypes of test data. There are at least three important sources for these differences: those due to inherent differences among machine learning methods, those attributable to different methods of feature selection, and finally, the use of different knowledge sources in training the various classifiers.

The work presented here evaluates and builds upon the Johns Hopkins University system (Yarowsky, Cucerzan, Florian, Schafer and Wicentowski 2001) that participated in the SENSEVAL2 exercise (Edmonds and Cotton 2001), wherein four classifiers (feature-enhanced Naïve Bayes, Cosine, bag-of-word Naïve Bayes and non-hierarchical Decision Lists) were used in combination to perform Word Sense Disambiguation (WSD). The approach was quite successful, obtaining state-of-the-art performance on the supervised tasks where it was applied (four lexical choice tasks – English, Spanish, Swedish and Basque, and two all-words tasks – Estonian and Czech). While the algorithms and techniques described in this paper are applicable to both types of word sense disambiguation tasks (*lexical-sample* and *all-words*), the current article focuses on the lexical-sample task.

This paper offers a detailed comparative evaluation and description of the problem of classifier combination over a structurally and procedurally diverse set of both well established and original classifiers: feature-enhanced Naïve Bayes, BayesRatio, Cosine, non-hierarchical Decision Lists, Transformation-Based Learning (TBL), and the MMVC classifiers, which will be briefly described in section 3. These systems have different space-searching strategies, ranging from discriminant functions (BayesRatio) to data likelihood (Bayes, Cosine) to decision rules (TBL, non-hierarchical Decision Lists), and are therefore amenable to classifier combination.

The article is organized as follows: section 2 outlines the rich space of features that were used by most classifiers; section 3 briefly describes the salient aspects of each classifier and any differences from their prototypical models; and section 4 describes and gives a detailed comparative analysis of the classifier combination experiments.

2 The feature space

The feature space is a critical part of classifier design, and its quality is often highly correlated with performance. For this reason, we used a rich feature space based on raw words, lemmas and Part-Of-Speech (POS) tags in a variety of positional and syntactic relationships to the target word. These positions include traditional bag-of-word context, local bigram and trigram collocations and several syntactic relationships based on predicate-argument structure. Their use is illustrated on a sample English sentence for the target word **train** in Table 1.

2.1 POS tagging and lemmatization

Part-of-speech tagger availability varied across the languages that we experimented upon. Transformation-based taggers (Ngai and Florian 2001) were trained on standard labeled data for English (Penn Treebank), Swedish (SUC-1 corpus), and Basque. For Spanish, a minimally-supervised tagger (Cucerzan and Yarowsky 2000) was used. Lemmatization was performed using trie-based supervised models for English, and a combination of supervised and unsupervised methods (Yarowsky and Wicentowski 2000) for all the other languages.

2.2 Syntactic features

The syntactic features extracted for a target word depend upon the target word's part of speech:

Table 1. *Example sentence and sample of extracted features*

Many mothers do not even try to toilet **train** their children until the age of 2 years or later ..

Feature type	Word	POS	Lemma	Feature type	Word	POS	Lemma
Context	...	...	...	*Syntactic (predicate-argument) features*			
Context	try	VB	try/N	Object	children	NNS	child/N
Context	to	TO	to/T	Prep	until	IN	until/I
Context	toilet	NN	toilet/N	ObjPrep	age	NN	age/N
Context	train	VBP	train/V	*Ngram collocational features*			
Context	their	DT	their/D	−1 bigram	toilet	NN	toilet/N
Context	children	NN	child/N	+1 bigram	their	DT	their/D
Context	...	...	...	−1/+1 trigram	to * their	TO-DT	to/T * their/D
Context	...	...	...	+1/+2 trigram	their children	DT-NN	their/D child/N

- for verbs: the head noun of the verb's object, particle/preposition and prepositional object;
- for nouns: the headword of any verb-object, subject-verb or noun-noun relationships identified for the target word; and
- for adjectives: the head noun modified by the adjective.

The extraction process is performed using heuristic patterns, implemented by regular expressions over the part-of-speech tags surrounding the target word. On the English data, the output from a text chunker (Ngai and Florian 2001) was also used as input to the regular expressions.

3 Classifier models for WSD

This section briefly introduces the classifier models used in the combination experiments. Among these models, the Naïve Bayes variants (henceforth NB) (Mooney 1996; Pedersen 1998; Manning and Schütze 1999; Yarowsky et al. 2001) and Cosine differ very little from off-the-shelf versions. Thus, only the differences will be described. The decision list models are based on the non-hierarchical variant described in Yarowsky (1996). The MMVC and TBL models are presented in Cucerzan and Yarowsky (2002) and Florian and Yarowsky (2002), respectively, and their framework is summarized here.

3.1 Vector-based models: extended Naïve Bayes and Cosine models

Many of the systems used in this research share a common vector representation, which captures traditional bag-of-words/lemmas, extended ngram and predicate-argument features in a single data structure. In these models, a vector is created for each document in the collection[1]: $\vec{d} = (d_j)_{j=1,|F|}$, $d_j = \frac{c_j}{N} W_j$, where F is the feature space, c_j is the number of times the feature f_j appears in document d, N is the number

[1] Given the focus of this work on lexical-sample task, we will denote as a 'document' the context of 1–5 sentences which contain the instance of the ambiguous word.

of words in d and W_j is a weight associated with the feature f_j.[2] Confusion between the same word participating in multiple feature roles is avoided by appending the feature values with their positional type (e.g. *children_object*, *toilet_L*, *their_R* are distinct from *children, toilet* and *their* in unmarked bag-of-words context).

The main difference between the extended models and others described in the literature, aside from the use of more sophisticated features, is the weighting of feature types. These differences yield a boost in the NB performance of between 3.5% (Basque) and 10% (Spanish), with an average improvement of 7.25% over the four languages.

3.2 The BayesRatio model

The *BayesRatio* model (henceforth BR) is a vector-based model using the likelihood ratio framework described in Gale, Church and Yarowsky (1992):

$$\begin{aligned}\hat{s} &= \arg\max_s \frac{P(s|d)}{P(\neg s|d)} = \arg\max_s \frac{P(s)\prod_{f\in F(d)} P(f|s)}{P(\neg s)\prod_{f\in F(d)} P(f|\neg s)} \\ &= \arg\max_s \left(\log\frac{P(s)}{P(\neg s)} + \sum_{f\in F(d)} \log\frac{P(f|s)}{P(f|\neg s)} \right)\end{aligned} \tag{1}$$

By utilizing a binary ratio for k-way modeling of feature probabilities, this approach performs well on tasks where data is sparse. While the BR model obtains roughly the same performance on the English WSD task as the NB system, BR outperforms NB on Spanish (by 1.1%) and Swedish (by 1.35%).

3.3 The mixture model

A direct mixture model that uses the same starting point as the algorithm presented in Walker (1987) was also investigated. Under this model, the conditional probability of a sense s given a target word in a context d is estimated as a mixture of the conditional sense probability distributions for individual context features:

$$P(s|d) = \sum_{f\in F(d)} P(s|f,d)P(f|d) \cong \sum_{f\in F(d)} P(s|f)P(f|d).$$

3.3.1 The Maximum Variance Correction (MVC) method

One problem arising from the sparseness of the training data is that the mixture model tends to overly favor the best represented senses in the training set. This problem is addressed by a second classification step that uses the variation of the components of the estimated posterior distributions across data $P(s|\cdot)$. The mean and variance of each component (for each s) of these distributions corresponding to

[2] The weight W_j depends on the type of the feature f_j: for the bag-of-words features, this weight is inversely proportional to the distance between the target word and the feature, while for predicate-argument and extended ngram features it is a predetermined weight which is set on a per-language basis.

the mixture model are computed by jackknife estimation on the training data. On test data, the classifications given by the mixture model are switched with the senses corresponding to the outliers with respect to the mean and variance estimated on training data when certain conditions are met, as described in Cucerzan and Yarowsky (2002).

The MVC method successfully addresses the problem of under-estimation of infrequent classes and significantly and consistently improves the performance of the mixture model across all languages (0.9–1.5% absolute value).

The joint model, called MMVC, proves to be a robust alternative to standard Bayesian models. With substantially different output to these models and the other statistical models investigated, MMVC is shown to be very suitable for the purpose of classifier combination.

3.4 TBL classifier

Transformation-Based Learning (TBL) is a general machine learning classification method described comprehensively by Brill (1995). It has been successfully applied to a variety of problems in natural language processing, including POS tagging, BaseNP identification, text chunking and prepositional phrase attachment.

To enable TBL to perform competitively in WSD, a number of enhancements to the basic algorithm were made:

- Support for bag-of-word type features was added, because they have an important effect on system performance in WSD.
- Redundancy was improved by adding to the list of rules, after the training phase has completed, all the rules that do not introduce an error. This modification has the effect of allowing for alternative redundant 'explanations' of the data that do not directly contribute to training data modeling, but may be seen alone in novel test data. This approach has been used in decision list modeling and yields additional robustness desirable in sparse data domains.
- Support for multiple correct classifications was added, because some of the samples in the WSD task have multiple gold-standard 'true' labels.

These modifications to the TBL algorithm resulted in statistically significant absolute performance improvements ranging from 1% on both English and Basque data to 2.2% on Spanish data.

The main advantages of TBL in WSD are its ability to identify strong clues in local and mid-range context and the fact that it is error-driven, while its main deficiency rests in its inability to aggregate information from multiple weak clues (e.g. long distance lexical relations).

4 WSD classifier combination: models and evaluation

Classifier combination has been theoretically and practically shown to be beneficial in terms of improving system accuracy. Perrone (1993) shows that, under the restrictive assumption that the n input classifiers are uncorrelated and have unbiased binary-output, the expected error is reduced by a factor of n when combining their classifications through averaging.

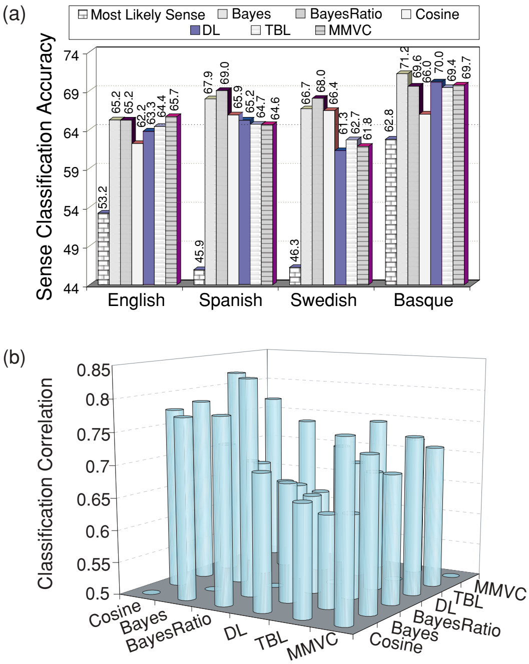

Fig. 1. Individual classifier properties: (a) Individual classifier performance, (b) Classifier inter-agreement.

System combination brings relatively little performance improvement over individual classifiers when the classifiers have a very high inter-agreement rate. Figure 1(b) shows the inter-agreement among the six classifiers presented in section 3, on the English data. Only two of them, Bayes and BayesRatio, have an agreement rate over 80% (measured using five-fold cross validation on training data), and pairwise agreement can be as low as 63%. The fact that the classifiers' behaviors are not strongly correlated means that the differences in performance among them can be systematically exploited to improve the overall classification. All individual classifiers have high stand-alone performance, as displayed in Figure 1(a), and their relative performance varies across languages, creating a good basis for classifier combination.

Unless it is explicitly stated otherwise, all of the following results were obtained by performing five-fold cross-validation on the Senseval2 training data. Where parameters needed to be estimated, a 3-1-1 split was used, training the systems on three parts, estimating parameters on the fourth (in a round-robin fashion) and testing on the fifth. Special care was taken so that no 'test' data was used in training classifiers or parameter estimation. The official Senseval2 test data was not touched until a single evaluation of our final frozen consensus system was performed shortly before submitting this article.

4.1 Combining classifier output

Most classifiers used in the experiments described here can output a probability distribution along with their classification: the Naïve Bayes, BayesRatio, Cosine and mixture models (described in Section 3) do it naturally, while TBL can be converted to output a (non-trivial) probabilistic classification (Florian et al. 2000). A general method of converting any classifier's output into a probabilistic one is presented in Duda, Hart and Stork (2000, Chapter 9.7).

Combining the probabilistic output of a set of classifiers $(C_i)_{i=1,n}$ which output a corresponding set of word sense probabilities $(P_i(s|d))_{i=1,n}$ can be described as a mixture of experts

$$P(s|d) = \frac{1}{Z}\sum_{i=1}^{n} f(P(s|d,i)) \cdot P(i|d) = \frac{1}{Z}\sum_{i=1}^{n} f(P_i(s|d)) \cdot P(i|d), \tag{2}$$

where i is the index of the ith classifier, $P(i|d)$ is a measure of how probable it is that the classifier C_i will have a correct output given document d, Z is a normalization factor, and f is a non-negative function. The role of function f is to provide a uniform specification for the models that are presented below:

- For a *count-based voting model*, the function f is

$$\sigma(P_i(s|d)) = \begin{cases} 1 & \text{if } s = \arg\max_{s'} P_i(s'|d) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

  and $P(i|d) = \frac{1}{n}$ in equation (2).

- For a *probability mixture model* (PM model), the function f will be the identity function (λ_i is the weight associated with the ith classifier):

$$P(s|d) = \sum_{i=1}^{n} P(i|d) \cdot P_i(s|d) = \sum_{i} \lambda_i \cdot P_i(s|d). \tag{4}$$

Ultimately, we are interested in obtaining a *hard* sense classification (i.e. each sample receives just one output label); therefore, the output of the combination classifier is obtained as

$$\hat{s} = \arg\max_{s} \sum_{i=1}^{n} f(P_i(s|d)) \cdot P(i|d) = \arg\max_{s} \sum_{i} f(P_i(s|d)) \cdot P(i|d). \tag{5}$$

4.1.1 Count-based and probability-based voting

One of the most popular ways of combining classifier output is by performing count-based voting (henceforth CBV): each system is allowed to vote for the sense that is most probable under its model; the sense that accumulates the largest number of votes wins. This algorithm is equivalent to finding the class $\hat{s}$ in equation (5), using the function $f = \sigma$, as defined in equation (4). This combination method improves classification accuracy by 1.3% on the English WSD data (as shown in Figure 2(a)), when compared with the single best system. Furthermore, similar improvement appears consistently across languages, as can be seen in Figure 2(b).

The CBV system's performance depends solely upon the performance of the hard-classification obtained from individual classifiers, and not on how well the

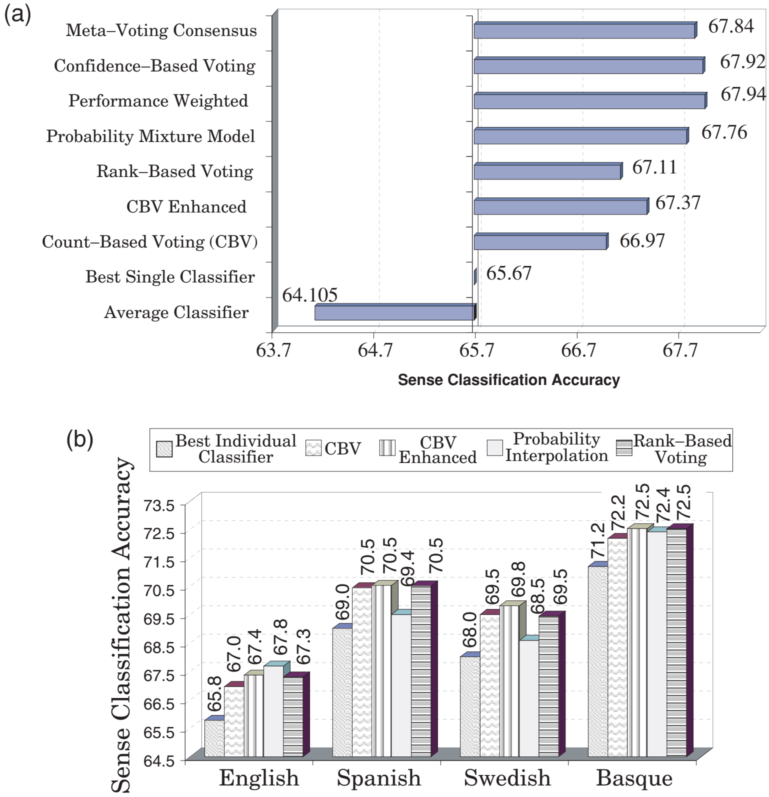

Fig. 2. Performance of the seven classifier combination models: (a) English lexical choice WSD performance, (b) WSD performance across four languages.

individual classifiers model the true sense probability distribution. This behavior has both good and bad implications: on the positive side, classifiers that by their nature make hard decisions (or estimate the output distribution poorly) can be added to the mixture without difficulty; on the negative side, there is no representation for close or uncertain decisions among senses that could be better modeled by incorporating sense probability distributions from individual classifiers. An effective way to address the latter issue, while still maintaining the advantages of the former, is to add a *mixture classifier* to the CBV system, based on the interpolation of the voting classifiers (we will call this classifier the *enhanced* CBV classifier). The classification decision (5) becomes in this case

$$\hat{s} = \arg\max_s \left(\sum_i \sigma(P_i(s|d)) + \lambda \cdot \sigma\left(\sum_i P_i(s|d) \right) \right) \tag{6}$$

where λ is a weight associated with the probability mixture model. This model assumes, as also do the CBV models, that each classifier is equally likely to produce a wrong classification, independent of the document to be classified: $P(i|d) = \frac{1}{n}$. Clearly, the probability mixture (henceforth PM) classifier can also be used as a stand-alone classifier.

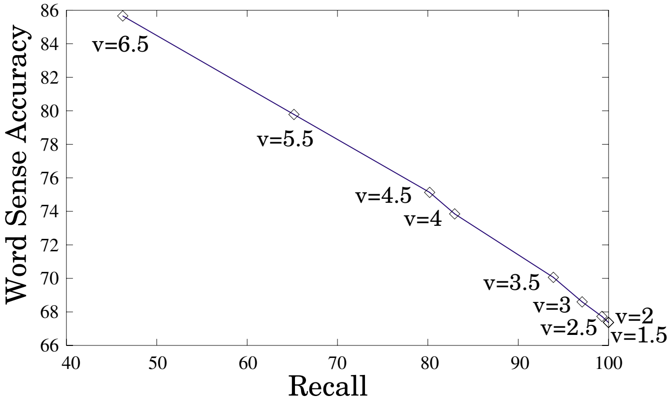

Fig. 3. Precision-recall curve for CBV: The displayed points are the voting counts that generate the particular (precision, recall) points.

Another probability-based combination method is the Rank-Based Voting (RBV) system, in which each classifier votes for every classification with a weight that is inversely proportional to the rank of the classification in the probabilistic output of the classifier.[3] The RBV classifier is less affected by poorly estimated probability distributions than the PM classifier, since it depends upon the relative ordering of the senses, and not the actual magnitudes of the probabilities.

Figure 2(b) compares the performance of the CBV methods with that of RBV and PM methods, across different languages. On English and Basque, the performance of the three combination methods (CBV enhanced, RBV and PM) is similar. It appears that the model probabilities are not as well estimated by all individual classifiers on Spanish and Swedish, because the PM model performs significantly worse (according to a paired McNemar test at a significance level of 10^{-6}). However, it should be noted that the PM combination is more amenable to adding further systems to the mix, i.e. it scales well with the number of component classifiers.

As a side-effect of the voting strategy, CBV classifiers receive a free and reasonably successful measure of confidence which they can use to restrict output only to their higher precision samples when desired. Confidence in a classification is measured directly by the number of votes that the classification received on the given sample. Figure 3 displays the precision-recall tradeoff of the CBV combination method applied to the five individual classifiers (the method's performance of 67.37% is presented in Figure 2(a)); the weight associated with the PM model in Equation (6) is 1.5 in this case. When the system is allowed to return only the samples on which it is most confident (i.e. the number of votes is 6.5), it obtains a precision of 85.7% at a recall level of 46%.

[3] The corresponding f function in equation (5) can be either $f(P_i(s|d)) = 1/rank_i(s)$ or $f(P_i(s|d)) = K - rank_i(s)$ (Borda counts), where $rank_i(s)$ is the number of senses that have a higher probability than S according to the probability distribution P_i and K is the number of senses.

4.1.2 Confidence-based combination

We explore an alternative regime under which to combine the outputs of multiple classifiers and present an effective method for estimating voting parameters $(P(i|d))_i$ in equation (5) from training data. This general approach is similar to the one presented in Ho, Hull and Srihari (1994), which describes a method of conditioning the parameters of classifier combination based on the degree of agreement among classifiers. The central idea is that when classifiers can output confidences which correlate positively with accuracy, the combination method should assign weights to classifiers proportional to their *combined* confidence on each sample. Voting parameters may vary as a function of the relative confidence of multiple systems in a joint model. This will allow classifiers to have their votes weighted differently in situations where many are confident, none is confident or only one is confident.

Computing bins for confidence

We first impose some simplifying constraints on the voting parameter space. To avoid excessive data fragmentation, we allow voting parameters to vary as a function of coarse-granularity confidence bins. In addition, although we could in theory explore an n-dimensional space of voting coefficients for our n-classifier system, we will allow only one voting parameter, λ, to vary. This is another way to minimize data fragmentation which can result in poor parameter estimation due to data sparseness. λ is defined to be the coefficient of the vote of a single system, the *varying* system C_1; the remaining $n-1$ classifiers, the *base* model, $C_2 \dots C_n$ are given unit votes. If the parameter in effect for a particular test sample j is denoted as $\lambda(j)$, the output of the final classifier combination is given by

$$\arg\max_{s\in S}\left(\lambda(j)\cdot\sigma(P_1(s|d_j))+\sum_{i=2}^{N}\sigma(P_i(s|d_j))\right). \tag{7}$$

Our experiments use the following joint confidence binning method: for both the *varying* system and the *base* system, the samples are ranked based on the system confidence function (the confidence of a set of classifiers is computed as the product of the individual classifiers' confidence). For each of the two systems, these ranked lists of samples are divided into 3 bins: the first bin consists of samples in the lowest 30% of the list sorted by confidence; the second bin encompasses those samples in the 30–70% range, and the final bin, the upper 70–100%. Samples are then assigned to a bin in the two-dimensional grid corresponding to the joint confidence. The voting coefficients λ_{xy} are the variables governing classifier combination, and were estimated by performing five-fold cross-validation on the training data.

We should also note here that restricting the combination to a single varying system and a base system composed of m unit voters has the effect of constraining the set of $[\lambda_{xy}]$ parameters which are capable of producing distinct outcomes. In particular, if the base system contains m unit voters then the distinct λ-ranges for each bin are

$$\lambda : \{0\},(0,1),\{1\},(1,2),\dots,(m-1,m),\{m\},(m,\infty) \tag{8}$$

Table 2. *Example of learned relative confidence bin voting parameters on a* 3×3 *grid*

		Varying system confidence		
		0–30%	30–70%	70–100%
Base	0–30%	$\lambda_{11} = 2.1$	$\lambda_{21} = 2.1$	$\lambda_{13} = 4.1$
System	30–70%	$\lambda_{12} = 2.1$	$\lambda_{22} = 2.1$	$\lambda_{23} = 2.1$
Confidence	70–100%	$\lambda_{31} = 0.0$	$\lambda_{32} = 0.0$	$\lambda_{33} = 0.1$

Table 3. *Classifier combination through voting, with and without joint-confidence binning, for English and Spanish. The English system consists of TBL as the varying system binned against Bayes, BayesRatio, Cosine, Decision List, and MMVC. The Spanish system uses Bayes as the varying system binned against TBL, BayesRatio, Decision List and Cosine. TBL is ideal for use as the varying system because of its low agreement with other classifiers. We chose to use Bayes for Spanish due to the relatively poor performance of TBL on the Spanish data*

	English	Spanish
Joint confidence bin voting	67.92	69.75
All-classifier CBV system	66.97	69.64
Performance Increase due to bining	**+0.95**	**+0.11**

There are $2m + 2$ such value ranges. However, since we are not interested in the (obfuscatory) results produced by allowing the varying system to participate in tied votes, we will only consider the distinct ranges $(0, 1), \ldots, (m - 1, m), (m, \infty)$.

Table 2 presents an example of voting parameters learned under a 3×3 binning scheme. We observe λ to be an increasing function of *varying* system confidence and a decreasing function of *base* system aggregate confidence, conforming to our intuitions.

Table 3 contrasts performance of confidence-binning systems for English and Spanish with the performance of the CBV systems consisting of the same classifiers. In the case of English, binning yields a substantial performance increase over the baseline CBV systems. For Spanish, where even small performance gains over the most accurate individual system (BayesRatio) are difficult to achieve, the difference between binning and a count-based voter is not as pronounced.

4.1.3 Performance-based combination

Another combination possibility is to use the estimated performance of a classifier to weight its contribution, specifically $P(i|d) = \Pr(C_i \text{ is correct})$ in Equation (5):

$$\hat{s} = \arg\max_{s \in S} \sum_{i=1}^{n} \Pr(C_i \text{ is correct}) \cdot P_i(s|d) \tag{9}$$

where the estimation of the performance is carried out via the above mentioned 3-1-1 split. The resulting accuracy of the performance-based combination is presented in Figure 2(a) for the English training data.

4.1.4 Meta-voting

As can be observed in Figure 2(b), there is no single best classifier across languages. On English and Basque the PM classifier performs the best. On Swedish and Spanish, the enhanced CBV and RBV classifiers obtain the best results. The classifier errors are positively correlated, and their biases can be used to obtain a classifier that performs close to the best performing system, by implementing count-based voting on the outputs of the k best performing classifiers overall. The combination is done using equation (4), with equal weights ($\lambda_i = \frac{1}{n}$). We call this classifier a *meta-voting* classifier, as its input is the output of other voting classifiers. The results obtained by using this classifier are also presented in Figure 2(a).

4.2 Individual classifiers' contribution to combination

One interesting issue related to classifier combination is the relative contribution of each classifier type to the overall performance. A useful measure of this contribution is the difference in performance between the combination system including a particular classifier and the combination system from which it is excluded. The more negative the difference in accuracy upon omission, the more valuable the individual classifier is to the ensemble system.

Figure 4(a) displays the drop in performance obtained by eliminating any particular classifier from the 6-way combination, across four languages, while Figure 4(b) shows the contribution of each classifier on the English data for different training sizes (10–80% of training data). Figure 4(b) is obtained by repeatedly sampling a prespecified ratio of training samples from three of the five cross-validation folds, and testing on the other two. Note that the classifiers with the greatest marginal contribution to the combined system performance are not always the best single performing classifiers (as can be observed by comparing Figure 1(a) with Figure 4(a)), but rather those with the most effective and *original* exploitation of the common feature space. On average, the classifier that contributes the most to the combined system's performance is the TBL classifier, with an average improvement of 0.66% across the four languages. Also, note that TBL and Decision List offer the greatest marginal contribution on smaller training sizes (as shown in Figure 4(b)).

4.3 Performance on test data

All the experiments presented in this paper so far have been based *strictly* on the original Senseval2 training sets using cross-validation. The Senseval2 test set was unused and unexamined to avoid even the possibility of indirect optimization on this official test data. As a final step, a single consensus system was created for each language using meta-voting on the top three classifier combination methods

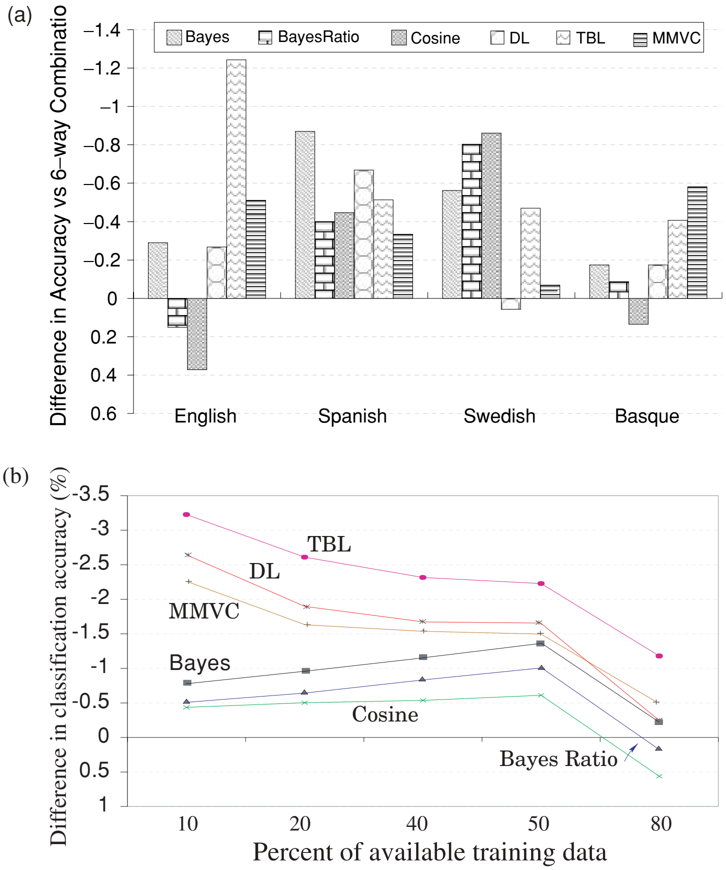

Fig. 4. Individual classifiers contribution to the voting performance: (a) Performance drop when eliminating one classifier (measure of marginal contribution), (b) Performance when eliminating one classifier, by training data size.

in Figure 2(b) for maximum robustness. To allow performance comparison with results from other sites, the frozen system for each language was applied once to the final SENSEVAL2 test sets shortly before submitting this article. Table 4 contrasts the performance obtained by the meta-voting (frozen) system to the average system performance and the official JHU system in the SENSEVAL2 exercise.[4] For each language, the new meta-voting system significantly outperforms all the participating systems. Also shown in Table 4 is the best performance of the individual classifiers (from section 4) in each language. The gain in performance introduced by the meta-voting combination is substantial. It is comparable to the standard deviation of system performance in the SENSEVAL2 evaluation, and the performance gain of Meta-combination relative to the best individual classifier is statistically significant for all four tested languages (by paired McNemar test) at $p \leq 0.01$.

[4] The official JHU submission is a CBV-enhanced system using Naïve Bayes classifiers (one based on lemmas and one based on words), cosine classifier and decision list classifiers.

Table 4. *Fine-grained performance on* Senseval2 *lexical-sample test data*

	English	Spanish	Swedish	Basque
Average system in Senseval2	55.7 ± 5.3%	59.6 ± 5.0%	58.4 ± 6.6%	74.4 ± 1.8%
JHU Senseval2 evaluation system	64.2%	71.2%	70.1%	75.7%
Best individual classifier	62.5%	69.6%	68.6%	75.2%
System meta-combination performance	66.3%	72.4%	71.9%	76.7%

5 Conclusion

In conclusion, we have presented a comparative evaluation study combining six structurally and procedurally different classifiers which utilize a rich common feature space. Various classifier combination methods, including count-based, confidence-based and probability-based combinations were described and evaluated. The experiments encompass supervised lexical sample tasks in four diverse languages: English, Spanish, Swedish, and Basque.

The experiments show substantial variation in single classifier performance across different languages and data sizes. They also show that this variation can be successfully exploited by five different classifier combination methods (and their meta-voting consensus), each of which outperforms both the single best classifier system and standard classifier combination models on each of the 4 focus languages. Furthermore, when these meta-voting consensus systems were frozen and applied once to the otherwise untouched Senseval2 test sets, they substantially outperformed the previously known Senseval2 results on each of the four languages.

References

Brill, E. and Wu, J. (1998) Classifier combination for improved lexical disambiguation. *Proceedings COLING-ACL'98*, pp. 191–195.

Brill, E. (1995) Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics* **21**(4): 543–565.

Cucerzan, S. and Yarowsky, D. (2000) Language independent minimally supervised induction of lexical probabilities. *Proceedings ACL-2000*, pp. 270–277. Hong Kong.

Cucerzan, S. and Yarowsky, D. (2002) Augmented mixture models for lexical disambiguation. *Proceedings EMNLP-2002*, pp. 33–40.

Duda, R. O., Hart, P. E. and Stork, D. G. (2001) *Pattern Classification.* Wiley-Interscience.

Edmonds, P. and Cotton, S. (2001) senseval-2: Overview. *Proceedings* senseval-2, pp. 1–6.

Florian, R. and Yarowsky, D. (2002) Modeling consensus: Classifier combination for word sense disambiguation. *Proceedings EMNLP'02*, pp. 25–32.

Florian, R., Henderson, J. C. and Ngai, G. (2000) Coaxing confidence from an old friend: Probabilistic classifications from transformation rule lists. *Proceedings EMNLP 2000*, pp. 26–34.

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

Ho, T. K., Hull, J. J. and Srihari, S. N. (1994) Decision combination in multiple classifier systems. *IEEE Trans. Pattern Analysis and Machine Intelligence* **16**(1): 66–75.

Kilgarriff, A. and Rosenzweig, J. (2000) Framework and results for English Senseval. *Comput. and the Humanities* **34**(1): 15–48.

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

Mooney, R. (1996) Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. *Proceedings EMNLP'96*, pp. 82–91.

Ngai, G and Florian, R. (2001) Transformation-based learning in the fast lane. *Proceedings NAACL*, pp. 40–47.

Pedersen, T. (1998) Naïve Bayes as a satisficing model. *Working Notes of the AAAI Spring Symposium on Satisficing Models.*

Pedersen, T. (2000) A simple approach to building ensembles of naive bayesian classifiers for word sense disambiguation. *Proceedings NAACL'00*, pp. 63–69.

Perrone, M. P. and Cooper, L. N. (1993) When networks disagree: Ensemble methods for hybrid neural networks. In: Mammone, R. J., editor, *Neural Networks for Speech and Image Processing*, pp. 126–142. Chapman & Hall.

Stevenson, M. and Wilks, Y. (2001) The interaction of knowledge sources in word sense disambiguation. *Computational Linguistics* **27**(3): 321–349.

van Halteren, H., Zavrel, J. and Daelemans, W. (1998) Improving data driven wordclass tagging by system combination. *Proceedings COLING-ACL'98*, pp. 491–497.

Walker, D. E. (1987) Knowledge resource tools for accessing large text files. In: Niremburg, S., editor, *Machine Translation: Theoretical and Methodological Issues*, pp. 247–261. Cambridge University Press.

Yarowsky, D. and Wicentowski, R. (2000) Minimally supervised morphological analysis by multimodal alignment. *Proceedings ACL-2000*, pp. 207–216. Hong Kong.

Yarowsky, D., Cucerzan, S., Florian, R., Schafer, C. and Wicentowski, R. (2001) The Johns Hopkins SENSEVAL2 system descriptions. *Proceedings SENSEVAL2*, pp. 163–166.

Yarowsky, D. (1996) Homograph disambiguation in speech synthesis. In: Olive, J., van Santen, J., Sproat, R. and Hirschberg, J., editors, *Progress in Speech Synthesis*, pp. 159–175. Springer-Verlag.