Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Philadelphia, July 2002, pp. 33-40. Association for Computational Linguistics.

Augmented Mixture Models for Lexical Disambiguation

Silviu Cucerzan and David Yarowsky
Department of Computer Science and
Center for Language and Speech Processing
Johns Hopkins University
Baltimore, MD 21218, USA
{silviu,yarowsky}@cs.jhu.edu

Abstract

This paper investigates several augmented mixture models that are competitive alternatives to standard Bayesian models and prove to be very suitable to word sense disambiguation and related classification tasks. We present a new classification correction technique that successfully addresses the problem of under-estimation of infrequent classes in the training data. We show that the mixture models are boosting-friendly and that both Adaboost and our original correction technique can improve the results of the raw model significantly, achieving state-of-the-art performance on several standard test sets in four languages. With substantially different output to Naïve Bayes and other statistical methods, the investigated models are also shown to be effective participants in classifier combination.

1 Introduction

The focus tasks of this paper are two related problems in lexical ambiguity resolution: Word Sense Disambiguation (WSD) and Context-Sensitive Spelling Correction (CSSC).

Word Sense Disambiguation has a long history as a computational task (Kelly and Stone, 1975), and the field has recently supported large-scale international system evaluation exercises in multiple languages (SENSEVAL-1, Kilgarriff and Palmer (2000), and SENSEVAL-2, Edmonds and Cotton (2001)).

General purpose Spelling Correction is also a long-standing task (e.g. McIlroy, 1982), traditionally focusing on resolving typographical errors such as transposition and deletion to find the closest "valid" word (in a dictionary or a morphological variant), typically ignoring context. Yet Kukich (1992) observed that about 25-50% of the spelling errors found in modern documents are either context-inappropriate misuses or substitutions of *valid* words (such as *principal* and *principle*) which are not detected by traditional spelling correctors. Previous work has addressed the problem of CSSC from a machine learning perspective, including Bayesian and Decision List models (Golding, 1995), Winnow (Golding and Roth, 1996) and Transformation-Based Learning (Mangu and Brill, 1997).

Generally, both tasks involve the selection between a relatively small set of alternatives per keyword (e.g. sense id's such as *church*/BUILDING and *church*/INSTITUTION or commonly confused spellings such as *quiet* and *quite*), and are dependent on local and long-distance collocational and syntactic patterns to resolve between the set of alternatives. Thus both tasks can share a common feature space, data representation and algorithm infrastructure. We present a framework of doing so, while investigating the use of mixture models in conjunction with a new error-correction technique as competitive alternatives to Bayesian models. While several authors have observed the fundamental similarities between CSSC and WSD (e.g. Berleant, 1995 and Roth, 1998), to our knowledge no previous comparative empirical study has tackled these two problems in a single unified framework.

2 Problem Formulation. Feature Space

The problem of lexical disambiguation can be modeled as a classification task, in which each instance of the word to be disambiguated (target word, henceforth), identified by its context, has to be labeled with one of the established sense labels $S = \{s_1, s_2, ..., s_n\}$.[1] The approaches we investigate are statistical methods $h : C \times S \rightarrow [0, 1]$, outputting conditional probability distributions over the sense set S given a context $c \in C$. The classification of a context c is generally made by choosing $argmax_{s \in S} h(c, s)$, but we also present an alterna-

[1] In the case of spelling correction, the classification labels are represented by the confusion set rather than sense labels (for example $S = \{then, than\}$).

... same table as the others but moved into the other **bar** with my pint and my ...			
Feature type	Word	POS	Lemma
Context features			
Context	moved/VBD	VBD	move/V
Context	into/IN	IN	into/I
Context	the/DT	DT	the/D
Context	other/JJ	JJ	other/J
Target	**bar**/NN	NN	**bar**/N
Context	with/IN	IN	with/I
Context	my/PRP$	PRP$	my/P
Context	pint/NN	NN	pint/N
Syntactic (predicate-argument) features			
ObjectTo	moved/VBD	VBD	move/V
Modifier	other/JJ	JJ	other/J
Bigram collocational features			
-1 Bigram	other/JJ	JJ	other/J
+1 Bigram	with/IN	IN	with/IN

Figure 1: Example context for WSD SENSEVAL-2 target word *bar* (inventory of 21 senses) and extracted features

... presents another {**piece,peace**} of the problem ...			
Feature type	Word	POS	Lemma
Context features			
Context	presents	VBZ	present/V
Context	another	DT	another/D
Target	{**peace,piece**}	NN	x/N
Context	of	IN	of/I
Context	the	DT	the/D
Context	problem	NN	problem/N
Syntactic (predicate-argument) features			
ObjectTo	presents	VBZ	present/V
Modifier	problem	NN	problem/N
Bigram collocational features			
-1 Bigram	another	DT	another/D
+1 Bigram	of	IN	of/I
Bigram POS environment			
POS-2-1	-	VBZ+DT	-
POS+1+2	-	IN+DT	-

Figure 2: Example context for the spelling confusion set {*piece,peace*} and extracted features

tive approach in Section 4.1.

The contexts C are represented as a collection of features. Previous work in WSD and CSSC (Golding, 1995; Bruce et al., 1996; Yarowsky, 1996; Golding and Roth, 1996; Pedersen, 1998) has found diverse feature types to be useful, including inflected words, lemmas and part-of-speech (POS) in a variety of collocational and syntactic relationships, including local bigrams and trigrams, predicate-argument relationships, and wide-context bag-of-words associations. Examples of the feature types we employ are illustrated in Figures 1 and 2.

The syntactic features are intended to capture the predicate-argument relationships in the syntactic window in which the target word occurs. Different relations are considered depending on the target word's POS. For nouns, these relations are: verb-object, subject-verb, modifier-noun, and noun-modified_noun; for verbs: verb-object, verb-particle/preposition, verb-prepositional_object; for adjectives: modifying_adjective-noun. Also, words with the same POS as the target word that are linked to the target word by coordinating conjunctions are extracted as sibling features. The extraction process is performed using simple heuristic patterns and regular expressions over the POS environment.

As Figure 2 shows, we considered for the CSSC task the POS bigrams of the immediate left and right word pairs as additional features in order to solve POS ambiguity and capture more of the syntactic environment in which the target word occurs (the elements of a confusion set often have disjoint or very different syntactic functions).

3 Mixture Models (MM)

We investigate in this Section a direct statistical model that uses the same starting point as the algorithm presented in Walker (1987). We then compare the functionality and the performance of this model to those of the widely used Naïve Bayes model for the WSD task (Gale et al., 1992; Mooney, 1996; Pedersen, 1998), enhanced with the full richer feature space beyond the traditional unordered bag-of-words.

Algorithm 1 Naïve Bayes Model

$$P(s|c) = \frac{P(s) \cdot P(c|s)}{P(c)} \cong \quad (1)$$

$$\frac{P(s) \cdot \prod_{w \in F(c)} P(w|s)}{\sum_{s' \in S} P(s') \cdot \prod_{w \in F(c)} P(w|s')} \quad (2)$$

It is known that Bayes decision rule is optimal if the distribution of the data of each class is known (Duda and Hart, 1973, ch. 2). However, the class-conditional distributions of the data are not known and have to be estimated. Both Naïve Bayes and the mixture model we investigated estimate $P(s|c)$ starting from mathematically correct formulations, and thus would be equivalent if the assumptions they make were correct. Naïve Bayes makes the assumption (used to transform Equation (1) into (2)) that the features are conditionally independent given a sense label. The mixture model makes a similar assumption, by regarding a document as being completely described by a union of independent features (Equation (3)). In practice, these are not true. Given the strong correlation and common redun-

dancy of the features in the case of WSD-related tasks, in conjunction with the limited training data on which the probabilities are estimated and the high dimensionality of the feature space, these assumptions lead to substantial modeling problems. Another important observation is that very many of the frequencies involved in the probability estimation are zero because of the very sparse feature space. Naïve Bayes depends heavily on probabilities not being zero and therefore it has to rely on smoothing. On the other hand, the mixture model is more robust to unseen events, without the need for explicit smoothing.

Under the proposed mixture model, the conditional probability of a sense s given a target word x in a context c is estimated as a mixture of the conditional sense probability distributions for individual context features:

Algorithm 2 Mixture Model

$$P(s|c) = \sum_{w \in F(c)} P(s|w,c) \cdot P(w|c) \cong \quad (3)$$

$$\sum_{w \in F(c)} P(s|w) \cdot P(w|c) \quad (4)$$

as opposed to the Naïve Bayes model in which the probability of a sense s given a context c is derived from the prior probability of s weighted by the conditional probabilities of the contextual features $F(c)$ given the sense.

The probabilities $P(s|w)$ in (4) and $P(w|s)$ in (2) can be computed as maximum likelihood estimates (MLE), by counting the co-occurrences of s and w versus the occurrences of w, respectively s in the training data. An extension to this classical estimation method is to use distance-weighted counts instead of raw counts for the relative frequencies:

$$P(s|w) = \frac{\Delta freq(w, T_{x|s})}{\Delta freq(w, T_x)} = \frac{\sum_{c_s \in T_{x|s}} \Delta(w, c_s)}{\sum_{c \in T_x} \Delta(w, c)} \quad (5)$$

$$P(w|s) = \frac{\Delta freq(w, T_{x|s})}{\sum_{w' \in F(c)} \Delta freq(w', T_{x|s})} \quad (6)$$

T_x denotes the training contexts of word x and $T_{x|s}$ the subset of T_x corresponding to sense s. When w is a syntactic headword, $\Delta(w,c)$ is computed by raw count. When w is a context word, $\Delta(w,c)$ is computed as a function of the position i of the target word x in c and the positions $j_1, ..., j_n$ where w occurs in c: $\Delta(w,c) = \sum_{l=1}^{n} \delta(i, j_l)$. If $\delta(i, j_l)$ are set to 1 regardless of the distance $|i - j_l|$ then MLE estimates are obtained. There are various other ways of choosing the weighting measure δ. One natural way is to transform the distance $|i - j_l|$ into a closeness measure by considering $\delta(i, j_l) = \frac{1}{1+|i-j_l|}$ (Manning and Schütze, 1999, ch. 14.1). This measure proves to be effective for the spelling correction task, where the words in the immediate vicinity are far more important than the rest of the context words[2], but imposes counterproductive differences between the much wider context positions (such as +30 vs. +31) used in WSD, especially when considering large context windows. Experimental results indicate that it is more effective to level out the local positional differences given by a continuous weighting, by instead using weight-equivalent regions which can be described with a simple step-function $\delta(j,t) = \frac{1}{1+\left\lfloor \frac{\sqrt{|j-t|}}{k} \right\rfloor}$, ($k$ is a constant[3]).

A filtering process based on the overall importance of a word w for the disambiguation of x is also employed, using alterations of the form $\frac{\Delta freq(w, T_{x|s})}{\Delta freq(w, T_x) + \alpha_{w,x}}$, with $\alpha_{w,x}$ proportional to the number of senses of target word x which it co-occurs with in the training set.[4] In this way, the words that occur only once in the training set, as well as those that occur with most of the senses of a word, providing no relevant information about the sense itself, are penalized.

Improvements obtained using weighted frequencies and filtering over MLE are shown in Table 1.

	Bayes	Mixture
MLE bag-of-words only	55.55	56.31
MLE with syntactic features	61.62	62.27
+ Weighting + Filtering	63.28	63.06
+ Collocational Senses[5]	65.70	65.41

Table 1: The increase in performance for successive variants of Bayes and Mixture Model as evaluated by 5-fold cross validation on SENSEVAL-2 English data

$P(w|c)$ can be seen as weighting factors in the mixture model formula (4). When w is a word,

[2]Golding and Schabes (1996) show that the most important words for CSSC are contained within a window of ±3.

[3]The results shown were obtained for $k = 2$ with term weights doubled within a ±3 context window. Various other functions and parameters values were tried on held-out parameter-optimization data for SENSEVAL-2.

[4]A normalization step is required to output probability distributions.

[5]The collocational sense information is specific to the SENSEVAL-2 English task and relies on the given inventory of collocation sense labels (e.g. *art_gallery%1:06:00::*).

$P(w|c)$ expresses the positional relationship between the occurrences of w and the target word x in c, and is computed using step-functions as described previously. When w is a syntactic headword, $P(w|c)$ is chosen as the average value of two ratios expressing the usefulness of the headword type for the given target word and respectively for the POS-class of the target word (adjective, noun, verb). These ratios are estimated by using a jackknife (hold-one-out) procedure on the training set and counting the number times the headword type is a good predictor versus the number of times it is a bad predictor.

Feature Type (position)	Value Lemma/POS	DMM $P(s\|w)$	Naïve Bayes $P(w\|s)$	δ
Syntactic Features				
SubjectTo	move/V	0	0	3
Modifier	other/J	0	0	8
Bigrams				
-1 Bigram	other/J	0	0	2
+1 Bigram	with/I	0.4444	0.0007	1
Contextual Features				
Context(-17)	pub/N	0.3677	0.0007	.3
Context(-13)	sit/V	0.5708	0.0028	.5
Context(-9)	table/N	0.7173	0.0008	.5
Context(-4)	move/V	0.2990	0.0007	1
Context(-3)	into/I	-	-	-
Context(-2)	the/D	-	-	-
Context(-1)	other/J	-	-	-
Target	**bar**/N	0.4296	[0.0530]	2
Context(+1)	with/I	-	-	-
Context(+2)	my/P	-	-	-
Context(+3)	pint/N	0.3333	0.0001	2
...	...	...	...	
Posterior probability $P(s\|c)$:		$\frac{1}{K}\sum$=.46	$\frac{P(s)}{Z}\prod$=.29	

Figure 3: A WSD example that shows the influence of syntactic, collocational and long-distance context features, the probability estimates used by Naïve Bayes and MM and their associated weights (δ), and the posterior probabilities of the true sense as computed by the two models.

As shown in Table 1, Bayes and mixture models yield comparable results for the given task. However, they capture the properties of the feature space in distinct ways (example applications of the two models on the sentence in Figure 1 are illustrated in Figure 3) and therefore, are very appropriate to be used together in combination (see Section 5.4).

4 Classification Correction and Boosting

We first present an original classification correction method based on the variation of posterior probability estimates across data and then the adaptation of the Adaboost method (Freund and Schapire, 1997) to the task of lexical classification.

4.1 The Maximum Variance Correction Method (MVC)

One problem arising from the sparseness of training data is that mixture models tend to excessively favor the best represented senses in the training set. A probable cause is that spurious words, which can not be considered general stopwords but do not carry sense-disambiguation information for a particular target word, may occur only by chance both in training and test data.[6] Another cause is the fact that mixture models search for decision surfaces linear in the feature space[7]; therefore, they can not make only correct classifications (unless the feature space can be divided by linear conditions) and the samples for the under-represented senses are likely to be interpreted as outliers.

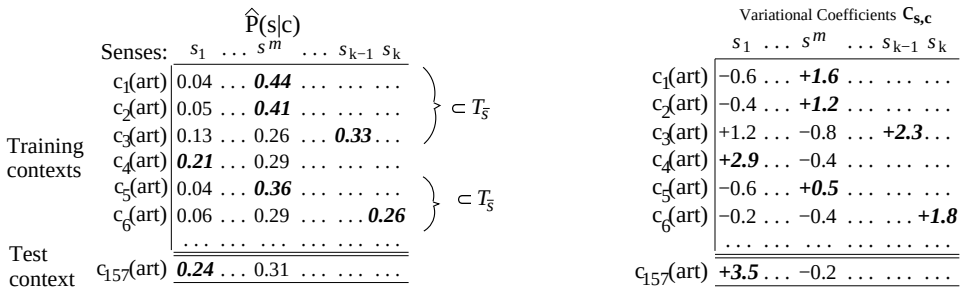
To address this estimation problem, a second classification step is employed, based on the observation that the deviation of a component of the posterior distribution from its expected value (as computed over the training set) can be as relevant as the maximum of the distribution $max_{s \in S}\widehat{P}(s|c)$. Instead of classifying each test context independently after estimating its sense probability distribution, we classify it by comparing it with the whole space of training contexts, for which the posterior distributions are computed using a jackknife procedure.

Figure 4(a) illustrates such an example: each line in the table represents the posterior distribution over senses given a context, each column contains the values corresponding to a particular sense in the posterior distributions of all contexts. Intuitively, sense s_1 may be preferred to the most likely sense s^m for the test context $c_{157}(art)$ despite the fact that the $\widehat{P}(s_1|c_{157})$ is smaller than $\widehat{P}(s^m|c_{157})$ because of the analogy with $c_4(art)$ and the "expected values" of the components corresponding to s_1 and s^m.

Unfortunately, we face again the problem of under-representation in the training data: the expected values in the posterior distributions for the under-represented senses when they express the correct classification can not be accurately estimated. Therefore, we have to look at the problem from another angle.

[6] For example, assuming that every context contains approximately the same number of such words, then given two senses, one represented in the training set by 20 examples, and the other one by 4, it is five times more likely that a spurious word in a test context co-occurs with the larger sampled sense.

[7] Roth (1998) shows that Bayes, TBL and Decision Lists also search for a decision surface which is a linear function in the feature space



(a) Probability distributions computed by MM using jackknife on the training set and a test context

(b) The variational coefficients for the example on the left

Figure 4: WSD example showing the utility of the MVC method. A sense s_1 with a high variational coefficient is preferred to the mode s^m of the MM distribution (the fields corresponding to the true sense are highlighted)

The mathematical support is provided by Chebyshev's inequality $P(|X - \mu| \geq \alpha\sigma) < \frac{1}{\alpha^2}$, which allows us to place an upper bound on the probability that the value of a random variable X is larger than a set value, given the mean μ and variance σ of X. Considering a finite selection $T = (x_i)_i$ from a distribution D for which μ and σ exist and can be estimated[8] as the empirical mean $\widehat{\mu} = \frac{1}{|T|}\sum_{x_i \in T} x_i$ and empirical variance $\widehat{\sigma}^2 = \frac{1}{|T|-1}\sum_{x_i \in T}(x_i - \widehat{\mu})^2$, and given another set $U = (y_k)_k$, the elements of U that are least probable as being generated from D are those for which the variational coefficients $vc_k = \frac{y_k - \widehat{\mu}}{\widehat{\sigma}}$ are large.

To apply this assumption to the disambiguation task, a set $T_{\bar{s}}$ containing the values $\widehat{P}(s|c)$ for all contexts c in the training set that are not labeled s is built for every sense s (see Figure 4(a)). In this way, the problem of poor representation of some senses is overcome and the selections $T_{\bar{s}}$ are large for all senses. An instance in the test set is considered more likely to correspond to a sense s if the estimated value $\widehat{P}(s|c)$ is an outlier with respect to $T_{\bar{s}}$ (see Figure 4(b)) and thus it is viewed as a candidate for having its classification changed to s.

Assuming that the selections $T_{\bar{s}}$ are representative and there exist first and second order moments for the underlying distributions (conditions which we call "good statistical properties"), an improvement in the accuracy $1 - \varphi$ of the classifier can be expected when choosing a sense with a variational coefficient $vc > \frac{1}{\sqrt{1-\varphi}}$ instead of the classifier distribution's mode $argmax_s \widehat{P}(s|c)$ (if such a sense exists). For example, knowing that the performance of the mixture model for SENSEVAL-2 is approximatively 0.65, the threshold for variational coefficients is set to 1.69. Because spurious words not only favor the better represented senses in the training set, but also can affect the variational coefficients of unlikely senses, some restrictions had to be imposed in our implementation to avoid the other extreme of favoring unlikely senses.

The mixture model does not guarantee the requirements imposed by the MVC method are met, but it has the advantage over the Bayesian model that each of the components of the posterior distribution it computes can be seen as a weighted mixture of random variables corresponding to the individual features. In the simplest case, when considering binary features, these variables are Bernoulli trials. Furthermore, if the trials have the same probability-mass function then a component of the posterior distribution will follow a binomial distribution, and therefore would have good statistical properties. In general, the underlying distributions can not be computed, but our experiments show that they usually have good statistical properties as required by MVC.

4.2 AdaBoost

AdaBoost is an iterative boosting algorithm introduced by Freund and Schapire (1997) shown to be successful for several natural language classification tasks. AdaBoost successively builds classifiers based on a weak learner (base learning algorithm) by weighting differently the examples in the training space, and outputs the final classification by mixing the predictions of the iteratively built classifiers. Because sense disambiguation is a multi-class problem, we chose to use version AdaBoost.M2.

We could not apply AdaBoost straightforwardly to the problem of sense disambiguation because of the high dimensionality and sparseness of the fea-

[8] It is hard to judge how well estimated these statistics are without making any distributional assumptions.

ture space. Superficial modeling of the training set can easily be achieved because of the singularity/rarity of many feature values in the context space, but this largely represents overfitting of the training data. In order to solve this problem, we use AdaBoost in conjunction with jackknife and a partial updating technique. At each round, N classifiers are built using as training all the examples in the training set except the one to be classified, and the weights are updated at feature level rather than context level. This modified Adaboost algorithm could only be implemented for the mixture model, which "perceives" the contexts as additive mixture of features. The Adaboost-enhanced mixture model is called AdaMixt henceforth.

5 Evaluation

We present a comparative study for four languages (English, Swedish, Spanish, and Basque) by performing 5-fold cross-validation on the SENSEVAL-2 lexical-sample training data, using the fine-grained sense inventory. For English and Swedish, for which POS-tagged training data was available to us, the fnTBL algorithm (Ngai and Florian, 2001) based on Brill (1995) was used to annotate the data, while for Spanish a mildly-supervised POS-tagging system similar to the one presented in Cucerzan and Yarowsky (2000) was employed. We also present the results obtained by the different algorithms on another WSD standard set, SENSEVAL-1, also by performing 5-fold cross validation on the original training data. For CSSC, we tested our system on the identical data from the Brown corpus used by Golding (1995), Golding and Roth (1996) and Mangu and Brill (1997). Finally, we present the results obtained by the investigated methods on a single run on the Senseval-1 and Senseval-2 test data.

The described models were initially trained and tested by performing 5-fold cross-validation on the SENSEVAL-2 English lexical-sample-task training data. When parameters needed to be estimated, jackknife or a 3-1-1 split (training and/or parameter estimation - testing) were used.

5.1 SENSEVAL-2

The English training set for SENSEVAL-2 is composed of 8861 instances representing 73 target words with an average number of 12.5 senses per word. Table 2 illustrates the performance of each of the studied models broken down by part-of-speech. As observed in most experiments, the feature-enhanced Naïve Bayes has the tendency to outperform by a small margin the raw mixture model, but because the latter proved to be boosting-friendly, its augmented versions achieved the highest final accuracies. The difference between MMVC and enhanced Naïve Bayes is significant (McNemar rejection risk of 4×10^{-3}).

	Adjectives	Nouns	Verbs	Overall
Most Likely	52.11	52.01	27.28	41.79
Naïve Bayes (FE)	73.18	72.74	55.54	65.70
Mixture	73.90	71.09	56.16	65.41
AdaMixt	**74.68**	72.17	56.41	66.09
MMVC	**74.68**	**73.06**	**57.06**	**66.72**

Table 2: Results using 5-fold cross validation on SENSEVAL-2 English lexical-sample training data

Figure 5 shows both the performance of the mixture model alone and in conjunction with MVC, and highlights the improvement in performance achieved by the latter for each of the 4 languages. All MMVC versus MM differences are statistically significant (for SENSEVAL-2 English data, the rejection probability of a paired McNemar test is 10^{-10}).

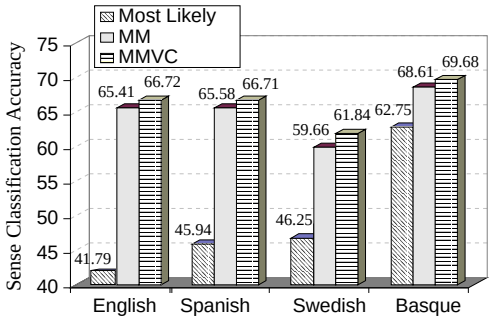


Figure 5: MM and MMVC performance by performing 5-fold cross validation on SENSEVAL-2 data for 4 languages

Figure 6 shows what is generally a log-linear increase in performance of MM alone and in combination with the MVC method over increasing training sizes. Because of the way the smallest training sets were created to include at least one example for each sense, they were more balanced as a side effect, and the compensations introduced by MVC were less productive as a result. Given more training data, MMVC starts to improve relative to the raw model both because the training sets become more unbalanced in their sense distributions and because the empirical moments and the variational coefficients on which the method relies are better estimated.

5.2 SENSEVAL-1

The systems used for SENSEVAL-2 English data were also evaluated on the SENSEVAL-1 training

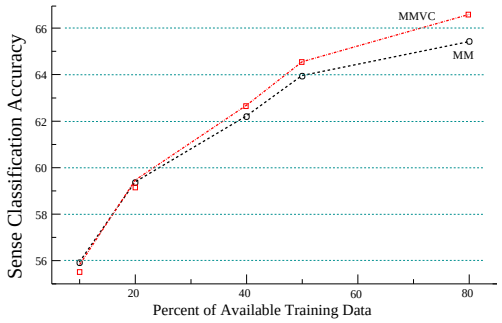


Figure 6: Learning Curve for MM and MMVC on SENSEVAL-2 English (cross-validated on heldout data)

data (30 words, 12479 instances, with an average of 10.8 senses per word) by using 5-fold cross validation. There was no further tuning of the feature space or model parameters to adapt them to the particularities of this new test set. Comparative performance is shown in Table 3. The difference between MMVC and enhanced Naïve Bayes is statistically significant (McNemar rejection risk 0.036).

	Adjectives	Nouns	Verbs	Overall
Most Likely	63.43	66.52	57.6	63.09
Naïve Bayes (FE)	75.67	84.15	76.65	80.16
Mixture	76.45	81.57	75.9	78.79
AdaMixt	76.83	83.39	**77.10**	80.16
MMVC	**78.49**	**84.79**	76.81	**81.06**

Table 3: Results using 5-fold cross validation on SENSEVAL-1 training data (English)

5.3 Spelling Correction

Both MM and the enhanced Bayes model obtain virtually the same overall performance[9] as the TriBayes system reported in (Golding and Schabes, 1996), which uses a similar feature space. The correction and boosting methods we investigated marginally improve the performance of the mixture model, as can be seen in Table 4 but they do not achieve the performance of RuleS 93.1% (Mangu and Brill, 1997) and Winnow 93.5% (Golding and Roth, 1996; Golding and Roth, 1999), methods that include features more directly specialized for spelling correction. Because of the small size of the test set, the differences in performance are due to only 14 and 20 more incorrectly classified examples respectively. More important than this difference[10] may be the fact that the systems built for WSD were able to achieve competitive performance with little to no adaptation (we only enriched the feature space by adding the POS bigrams to the left and right of the target word and changed the weighting model as presented in Section 3 because spelling correction relies more on the immediate than long-distance context). Another important aspect that can be seen in Table 4 is that there was no model that constantly performed best in all situations, suggesting the advantage of developing a diverse space of models for classifier combination.

[9] All figures reported are for the standard 14 confusion sets; the accuracies for the 18 sets are generally higher.

[10] We did not have the actual classifications from the other systems to check the significance of the difference.

	test size	M.L.	Bayes	MM	AdaMixt	MMVC
accept	50	70.0	92.0	90.0	90.0	**94.2**
affect	49	91.8	95.9	**98.0**	**98.0**	93.9
among	186	71.5	80.6	78.5	**81.2**	80.6
amount	123	71.5	79.7	79.7	82.9	**83.7**
begin	146	93.2	96.6	96.6	**97.3**	96.6
country	62	91.9	93.5	**95.2**	93.5	93.5
lead	49	46.9	93.9	91.8	**95.9**	91.8
past	74	68.9	86.5	**93.2**	**93.2**	**93.2**
peace	50	44.0	78.0	**80.0**	78.0	**80.0**
principal	34	58.8	82.3	**88.2**	85.3	**88.2**
quiet	66	83.3	93.9	93.9	93.9	**95.5**
raise	39	64.1	**87.2**	84.6	84.6	**87.2**
than	514	63.4	**96.9**	96.5	96.5	96.5
weather	61	86.9	**98.4**	95.1	96.7	**98.4**
Overall	1503	71.1	91.2	91.2	91.8	**92.2**

Table 4: Results on the standard 14 CSSC data sets

5.4 Using MMVC in Classifier Combination

The investigated MMVC model proves to be a very effective participant in classifier combination, with substantially different output to Naïve Bayes (9.6% averaged complementary rate, as defined in Brill and Wu (1998)). Table 5 shows the improvement obtained by adding the MMVC model to empirically the best voting system we had using Bayes, BayesRatio, TBL and Decision Lists (all classifier combination methods tried and their results are presented exhaustively in Florian and Yarowsky (2002)). The improvement is significant in both cases, as measured by a paired McNemar test: 1.7×10^{-7} for SENSEVAL-1 data, 1.8×10^{-7} for SENSEVAL-2 data.

	without MMVC	with MMVC	error reduction
Senseval1	82.26	**83.06**	4.5%
Senseval2	67.53	**68.66**	3.5%

Table 5: The contribution of MMVC in a rank-based classifier combination on SENSEVAL-1 and SENSEVAL-2 English as computed by 5-fold cross validation over training data

MMVC is also the top performer of the 5 systems mentioned above on SENSEVAL-2 English test

data, with an accuracy of 62.5%. Table 6 contrasts the performance obtained by the MMVC method to the average and best system performance in the two SENSEVAL exercises.

SENSEVAL-1 (30 target words, 7446 instances)

Average / Best SENSEVAL-1 Competitor	73.1 ± 2.9 / 77.1
MMVC alone	76.9
Classifier combination with MMVC	**80.0**

SENSEVAL-2 (73 target words, 4328 instances)

Average / Best SENSEVAL-2 Competitor	55.7 ± 5.3 / 64.2
MMVC alone	62.5
Classifier combination with MMVC	**66.5**

Table 6: Accuracy on SENSEVAL-1 and SENSEVAL-2 English test data (only the supervised systems with a coverage of at least 97% were used to compute the mean and variance)

6 Conclusion

We investigated the properties and performance of mixture models and two augmenting methods in an unified framework for Word Sense Disambiguation and Context-Sensitive Spelling Correction, showing experimentally that such joint models can successfully match and exceed the performance of feature-enhanced Bayesian models. The new classification correction method (MVC) we propose successfully addresses the problem of under-estimation of less likely classes, consistently and significantly improving the performance of the main mixture model across all tasks and languages. Finally, since the mixture model and its improvements performed well on two major tasks and several multilingual data sets, we believe that they can be productively applied to other related high-dimensionality lexical classification problems, including named-entity classification, topic classification, and lexical choice in machine translation.

References

D. Berleant. 1995. Engineering "word experts" for word disambiguation. *Natural Language Engineering*, 1(4):339–362.

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

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

R. Bruce, J. Wiebe, and T. Pedersen. 1996. The measure of a model. In *Proceedings of EMNLP-1996*, pages 101–112.

S. Cucerzan and D. Yarowsky. 2000. Language independent minimally supervised induction of lexical probabilities. In *Proceedings of ACL-2000*, pages 270–277.

R. O. Duda and P. E. Hart. 1973. *Pattern Classification and Scene Analysis*. Wiley.

P. Edmonds and S. Cotton. 2001. SENSEVAL-2 overview. In *Proceedings of* SENSEVAL*-2*, pages 1–6.

R. Florian and D. Yarowsky. 2002. Modeling consensus: Classifier combination for word sense disambiguation. In *Proceedings of EMNLP-2002*.

Y. Freund and R. E. Schapire. 1997. A decision-theoretic generalization of on-line learning and application to boosting. *Journal of Computer and System Sciences*, 55:119–139.

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

A. R. Golding and D. Roth. 1996. Applying winnow to context-sensitive spelling correction. In *Machine Learning: Proceedings of the 13th International Conference*, pages 182–190.

A. R. Golding and D. Roth. 1999. A winnow-based approach to context-sensitive spelling correction. *Machine Learning*, 34(1-3):107–130.

A. R. Golding and Y. Schabes. 1996. Combining trigram-based and feature-based methods for context-sensitive spelling correction. In *Proceedings of ACL-1996*, pages 71–78.

A. R. Golding. 1995. A Bayesian hybrid method for context-sensitive spelling correction. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 39–53.

E. F. Kelly and P. J. Stone. 1975. *Computer Recognition of English Word Senses*. North Holland Press.

A. Kilgarriff and M. Palmer. 2000. Introduction to the special issue on SENSEVAL. *Computers and the Humanities*, 34(1-2):1–13.

K. Kukich. 1992. Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24(4):377–439.

L. Mangu and E. Brill. 1997. Automatic rule acquisition for spelling correction. In *Proceedings of the 14th International Conference on Machine Learning*, pages 734–741.

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

M. D. McIlroy. 1982. Development of a spelling list. *j-IEEE-TRANS-COMM*, COM-30(1):91–99.

R. J. Mooney. 1996. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In *Proceedings of EMNLP-1996*, pages 82–91.

G. Ngai and R. Florian. 2001. Transformation-based learning in the fast lane. In *Proceedings of NAACL-2001*, pages 40–47.

T. Pedersen. 1998. Naïve Bayes as a satisficing model. In *Working Notes of the AAAI Spring Symposium on Satisficing Models*, pages 60–67.

D. Roth. 1998. Learning to resolve natural language ambiguities: a unified approach. In *Proceedings of the 15th Conference of the AAAI*, pages 806–813.

D. E. Walker. 1987. Knowledge resource tools for accessing large text files. In Sergei Nirenburg, editor, *Machine Translation: Theoretical and Methodological Issues*, pages 247–261. Cambridge University Press.

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