

Estimating Class Priors in Domain Adaptation for Word Sense Disambiguation

Yee Seng Chan and Hwee Tou Ng
Department of Computer Science
National University of Singapore
3 Science Drive 2, Singapore 117543
{chanys, nght}@comp.nus.edu.sg

Abstract

Instances of a word drawn from different domains may have different sense priors (the proportions of the different senses of a word). This in turn affects the accuracy of word sense disambiguation (WSD) systems trained and applied on different domains. This paper presents a method to estimate the sense priors of words drawn from a new domain, and highlights the importance of using well calibrated probabilities when performing these estimations. By using well calibrated probabilities, we are able to estimate the sense priors effectively to achieve significant improvements in WSD accuracy.

1 Introduction

Many words have multiple meanings, and the process of identifying the correct meaning, or sense of a word in context, is known as word sense disambiguation (WSD). Among the various approaches to WSD, corpus-based supervised machine learning methods have been the most successful to date. With this approach, one would need to obtain a corpus in which each ambiguous word has been manually annotated with the correct sense, to serve as training data.

However, supervised WSD systems faced an important issue of domain dependence when using such a corpus-based approach. To investigate this, Escudero et al. (2000) conducted experiments using the DSO corpus, which contains sentences drawn from two different corpora, namely Brown Corpus (BC) and Wall Street Journal (WSJ). They found that training a WSD system on one part (BC or WSJ) of the DSO corpus and applying it to the

other part can result in an accuracy drop of 12% to 19%. One reason for this is the difference in sense priors (i.e., the proportions of the different senses of a word) between BC and WSJ. For instance, the noun *interest* has these 6 senses in the DSO corpus: sense 1, 2, 3, 4, 5, and 8. In the BC part of the DSO corpus, these senses occur with the proportions: 34%, 9%, 16%, 14%, 12%, and 15%. However, in the WSJ part of the DSO corpus, the proportions are different: 13%, 4%, 3%, 56%, 22%, and 2%. When the authors assumed they knew the sense priors of each word in BC and WSJ, and adjusted these two datasets such that the proportions of the different senses of each word were the same between BC and WSJ, accuracy improved by 9%. In another work, Agirre and Martinez (2004) trained a WSD system on data which was automatically gathered from the Internet. The authors reported a 14% improvement in accuracy if they have an accurate estimate of the sense priors in the evaluation data and sampled their training data according to these sense priors. The work of these researchers showed that when the domain of the training data differs from the domain of the data on which the system is applied, there will be a decrease in WSD accuracy.

To build WSD systems that are portable across different domains, estimation of the sense priors (i.e., determining the proportions of the different senses of a word) occurring in a text corpus drawn from a domain is important. McCarthy et al. (2004) provided a partial solution by describing a method to predict the predominant sense, or the most frequent sense, of a word in a corpus. Using the noun *interest* as an example, their method will try to predict that sense 1 is the predominant sense in the BC part of the DSO corpus, while sense 4 is the predominant sense in the WSJ part of the

corpus.

In our recent work (Chan and Ng, 2005b), we directly addressed the problem by applying machine learning methods to automatically estimate the sense priors in the target domain. For instance, given the noun *interest* and the WSJ part of the DSO corpus, we attempt to estimate the proportion of each sense of *interest* occurring in WSJ and showed that these estimates help to improve WSD accuracy. In our work, we used naive Bayes as the training algorithm to provide posterior probabilities, or class membership estimates, for the instances in the target domain. These probabilities were then used by the machine learning methods to estimate the sense priors of each word in the target domain.

However, it is known that the posterior probabilities assigned by naive Bayes are not reliable, or not well calibrated (Domingos and Pazzani, 1996). These probabilities are typically too extreme, often being very near 0 or 1. Since these probabilities are used in estimating the sense priors, it is important that they are well calibrated.

In this paper, we explore the estimation of sense priors by first calibrating the probabilities from naive Bayes. We also propose using probabilities from another algorithm (logistic regression, which already gives well calibrated probabilities) to estimate the sense priors. We show that by using well calibrated probabilities, we can estimate the sense priors more effectively. Using these estimates improves WSD accuracy and we achieve results that are significantly better than using our earlier approach described in (Chan and Ng, 2005b).

In the following section, we describe the algorithm to estimate the sense priors. Then, we describe the notion of being well calibrated and discuss why using well calibrated probabilities helps in estimating the sense priors. Next, we describe an algorithm to calibrate the probability estimates from naive Bayes. Then, we discuss the corpora and the set of words we use for our experiments before presenting our experimental results. Next, we propose using the well calibrated probabilities of logistic regression to estimate the sense priors, and perform significance tests to compare our various results before concluding.

2 Estimation of Priors

To estimate the sense priors, or a priori probabilities of the different senses in a new dataset,

we used a confusion matrix algorithm (Vucetic and Obradovic, 2001) and an EM based algorithm (Saerens et al., 2002) in (Chan and Ng, 2005b). Our results in (Chan and Ng, 2005b) indicate that the EM based algorithm is effective in estimating the sense priors and achieves greater improvements in WSD accuracy compared to the confusion matrix algorithm. Hence, to estimate the sense priors in our current work, we use the EM based algorithm, which we describe in this section.

2.1 EM Based Algorithm

Most of this section is based on (Saerens et al., 2002). Assume we have a set of labeled data D_L with n classes and a set of N independent instances $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ from a new data set. The likelihood of these N instances can be defined as:

$$\begin{aligned} L(\mathbf{x}_1, \dots, \mathbf{x}_N) &= \prod_{k=1}^N p(\mathbf{x}_k) \\ &= \prod_{k=1}^N \left[\sum_{i=1}^n p(\mathbf{x}_k, \omega_i) \right] \\ &= \prod_{k=1}^N \left[\sum_{i=1}^n p(\mathbf{x}_k|\omega_i)p(\omega_i) \right] \quad (1) \end{aligned}$$

Assuming the within-class densities $p(\mathbf{x}_k|\omega_i)$, i.e., the probabilities of observing \mathbf{x}_k given the class ω_i , do not change from the training set D_L to the new data set, we can define: $p(\mathbf{x}_k|\omega_i) = p_L(\mathbf{x}_k|\omega_i)$. To determine the a priori probability estimates $\hat{p}(\omega_i)$ of the new data set that will maximize the likelihood of (1) with respect to $p(\omega_i)$, we can apply the iterative procedure of the EM algorithm. In effect, through maximizing the likelihood of (1), we obtain the a priori probability estimates as a by-product.

Let us now define some notations. When we apply a classifier trained on D_L on an instance \mathbf{x}_k drawn from the new data set D_U , we get $\hat{p}_L(\omega_i|\mathbf{x}_k)$, which we define as the probability of instance \mathbf{x}_k being classified as class ω_i by the classifier trained on D_L . Further, let us define $\hat{p}_L(\omega_i)$ as the a priori probabilities of class ω_i in D_L . This can be estimated by the class frequency of ω_i in D_L . We also define $\hat{p}^{(s)}(\omega_i)$ and $\hat{p}^{(s)}(\omega_i|\mathbf{x}_k)$ as estimates of the new a priori and a posteriori probabilities at step s of the iterative EM procedure. Assuming we initialize $\hat{p}^{(0)}(\omega_i) = \hat{p}_L(\omega_i)$, then for each instance \mathbf{x}_k in D_U and each class ω_i , the EM

algorithm provides the following iterative steps:

$$\hat{p}^{(s)}(\omega_i|\mathbf{x}_k) = \frac{\hat{p}_L(\omega_i|\mathbf{x}_k) \frac{\hat{p}^{(s)}(\omega_i)}{\hat{p}_L(\omega_i)}}{\sum_{j=1}^n \hat{p}_L(\omega_j|\mathbf{x}_k) \frac{\hat{p}^{(s)}(\omega_j)}{\hat{p}_L(\omega_j)}} \quad (2)$$

$$\hat{p}^{(s+1)}(\omega_i) = \frac{1}{N} \sum_{k=1}^N \hat{p}^{(s)}(\omega_i|\mathbf{x}_k) \quad (3)$$

where Equation (2) represents the expectation E-step, Equation (3) represents the maximization M-step, and N represents the number of instances in D_U . Note that the probabilities $\hat{p}_L(\omega_i|\mathbf{x}_k)$ and $\hat{p}_L(\omega_i)$ in Equation (2) will stay the same throughout the iterations for each particular instance \mathbf{x}_k and class ω_i . The new a posteriori probabilities $\hat{p}^{(s)}(\omega_i|\mathbf{x}_k)$ at step s in Equation (2) are simply the a posteriori probabilities in the conditions of the labeled data, $\hat{p}_L(\omega_i|\mathbf{x}_k)$, weighted by the ratio of the new priors $\hat{p}^{(s)}(\omega_i)$ to the old priors $\hat{p}_L(\omega_i)$. The denominator in Equation (2) is simply a normalizing factor.

The a posteriori $\hat{p}^{(s)}(\omega_i|\mathbf{x}_k)$ and a priori probabilities $\hat{p}^{(s)}(\omega_i)$ are re-estimated sequentially during each iteration s for each new instance \mathbf{x}_k and each class ω_i , until the convergence of the estimated probabilities $\hat{p}^{(s)}(\omega_i)$. This iterative procedure will increase the likelihood of (1) at each step.

2.2 Using A Priori Estimates

If a classifier estimates posterior class probabilities $\hat{p}_L(\omega_i|\mathbf{x}_k)$ when presented with a new instance \mathbf{x}_k from D_U , it can be directly adjusted according to estimated a priori probabilities $\hat{p}(\omega_i)$ on D_U :

$$\hat{p}_{adjust}(\omega_i|\mathbf{x}_k) = \frac{\hat{p}_L(\omega_i|\mathbf{x}_k) \frac{\hat{p}(\omega_i)}{\hat{p}_L(\omega_i)}}{\sum_j \hat{p}_L(\omega_j|\mathbf{x}_k) \frac{\hat{p}(\omega_j)}{\hat{p}_L(\omega_j)}} \quad (4)$$

where $\hat{p}_L(\omega_i)$ denotes the a priori probability of class ω_i from D_L and $\hat{p}_{adjust}(\omega_i|\mathbf{x}_k)$ denotes the adjusted predictions.

3 Calibration of Probabilities

In our earlier work (Chan and Ng, 2005b), the posterior probabilities assigned by a naive Bayes classifier are used by the EM procedure described in the previous section to estimate the sense priors $\hat{p}(\omega_i)$ in a new dataset. However, it is known that the posterior probabilities assigned by naive Bayes are not well calibrated (Domingos and Pazani, 1996).

It is important to use an algorithm which gives well calibrated probabilities, if we are to use the probabilities in estimating the sense priors. In this section, we will first describe the notion of being well calibrated before discussing why having well calibrated probabilities helps in estimating the sense priors. Finally, we will introduce a method used to calibrate the probabilities from naive Bayes.

3.1 Well Calibrated Probabilities

Assume for each instance \mathbf{x} , a classifier outputs a probability $S_{\omega_i}(\mathbf{x})$ between 0 and 1, of \mathbf{x} belonging to class ω_i . The classifier is well-calibrated if the empirical class membership probability $p(\omega_i|S_{\omega_i}(\mathbf{x}) = t)$ converges to the probability value $S_{\omega_i}(\mathbf{x}) = t$ as the number of examples classified goes to infinity (Zadrozny and Elkan, 2002). Intuitively, if we consider all the instances to which the classifier assigns a probability $S_{\omega_i}(\mathbf{x})$ of say 0.6, then 60% of these instances should be members of class ω_i .

3.2 Being Well Calibrated Helps Estimation

To see why using an algorithm which gives well calibrated probabilities helps in estimating the sense priors, let us rewrite Equation (3), the M-step of the EM procedure, as the following:

$$\hat{p}^{(s+1)}(\omega_i) = \frac{1}{N} \sum_{t \in S_{\omega_i}} \sum_{k \in \{q: S_{\omega_i}(\mathbf{x}_q) = t\}} \hat{p}^{(s)}(\omega_i|\mathbf{x}_k) \quad (5)$$

where $S_{\omega_i} = \{t_1, \dots, t_m\}$ denotes the set of posterior probability values for class ω_i , and $S_{\omega_i}(\mathbf{x}_q)$ denotes the posterior probability of class ω_i assigned by the classifier for instance \mathbf{x}_q .

Based on t_1, \dots, t_m , we can imagine that we have m bins, where each bin is associated with a specific t value. Now, distribute all the instances in the new dataset D_U into the m bins according to their posterior probabilities $S_{\omega_i}(\mathbf{x})$. Let B_l , for $l = 1, \dots, m$, denote the set of instances in bin l .

Note that $|B_1| + \dots + |B_l| + \dots + |B_m| = N$. Now, let p_l denote the proportion of instances with true class label ω_i in B_l . Given a well calibrated algorithm, $p_l = t_l$ by definition and Equation (5) can be rewritten as:

$$\begin{aligned} \hat{p}^{(s+1)}(\omega_i) &= \frac{1}{N} (t_1|B_1| + \dots + t_m|B_m|) \\ &= \frac{1}{N} (p_1|B_1| + \dots + p_m|B_m|) \\ &= \frac{N_{\omega_i}}{N} \end{aligned} \quad (6)$$

Input: training set (p_k, y_k) sorted in ascending order of p_k
Initialize $g_k = y_k$
While $\exists k$ such that $g_j, \dots, g_{k-1} > g_k, \dots, g_l$, where
 $g_j = \dots = g_{k-1}$ and $g_k = \dots = g_l$ ($j < k \leq l$)
Set $m = \frac{\sum_{q=j}^l g_q}{l-j+1}$
Replace g_j, \dots, g_l with m

Figure 1: PAV algorithm.

where N_{ω_i} denotes the number of instances in D_U with true class label ω_i . Therefore, $\hat{p}^{(s+1)}(\omega_i)$ reflects the proportion of instances in D_U with true class label ω_i . Hence, using an algorithm which gives well calibrated probabilities helps in the estimation of sense priors.

3.3 Isotonic Regression

Zadrozny and Elkan (2002) successfully used a method based on isotonic regression (Robertson et al., 1988) to calibrate the probability estimates from naive Bayes. To compute the isotonic regression, they used the pair-adjacent violators (PAV) (Ayer et al., 1955) algorithm, which we show in Figure 1. Briefly, what PAV does is to initially view each data value as a level set. While there are two adjacent sets that are out of order (i.e., the left level set is above the right one) then the sets are combined and the mean of the data values becomes the value of the new level set.

PAV works on binary class problems. In a binary class problem, we have a positive class and a negative class. Now, let $D = (p_k, \mathbf{x}_k)$, $1 \leq k \leq N$, where $\mathbf{x}_1, \dots, \mathbf{x}_N$ represent N examples and p_k is the probability of \mathbf{x}_k belonging to the positive class, as predicted by a classifier. Further, let y_k represent the true label of \mathbf{x}_k . For a binary class problem, we let $y_k = 1$ if \mathbf{x}_k is a positive example and $y_k = 0$ if \mathbf{x}_k is a negative example. The PAV algorithm takes in a set of (p_k, y_k) , sorted in ascending order of p_k and returns a series of increasing step-values, where each step-value $g_{j,l}$ (denoted by m in Figure 1) is associated with a lowest boundary value p_j and a highest boundary value p_l . We performed 10-fold cross-validation on the training data to assign values to p_k . We then applied the PAV algorithm to obtain values for g_k . To obtain the calibrated probability estimate for a test instance \mathbf{x} , we find the boundary values p_j and p_l where $p_j \leq S_{\omega_i}(\mathbf{x}) \leq p_l$ and assign $g_{j,l}$ as the calibrated probability estimate.

To apply PAV on a multiclass problem, we first reduce the problem into a number of binary class

problems. For reducing a multiclass problem into a set of binary class problems, experiments in (Zadrozny and Elkan, 2002) suggest that the one-against-all approach works well. In one-against-all, a separate classifier is trained for each class ω_i , where examples belonging to class ω_i are treated as positive examples and all other examples are treated as negative examples. A separate classifier is then learnt for each binary class problem and the probability estimates from each classifier are calibrated. Finally, the calibrated binary-class probability estimates are combined to obtain multiclass probabilities, computed by a simple normalization of the calibrated estimates from each binary classifier, as suggested by Zadrozny and Elkan (2002).

4 Selection of Dataset

In this section, we discuss the motivations in choosing the particular corpora and the set of words used in our experiments.

4.1 DSO Corpus

The DSO corpus (Ng and Lee, 1996) contains 192,800 annotated examples for 121 nouns and 70 verbs, drawn from BC and WSJ. BC was built as a balanced corpus and contains texts in various categories such as religion, fiction, etc. In contrast, the focus of the WSJ corpus is on financial and business news. Escudero et al. (2000) exploited the difference in coverage between these two corpora to separate the DSO corpus into its BC and WSJ parts for investigating the domain dependence of several WSD algorithms. Following their setup, we also use the DSO corpus in our experiments.

The widely used SEMCOR (SC) corpus (Miller et al., 1994) is one of the few currently available manually sense-annotated corpora for WSD. SEMCOR is a subset of BC. Since BC is a balanced corpus, and training a classifier on a general corpus before applying it to a more specific corpus is a natural scenario, we will use examples from BC as training data, and examples from WSJ as evaluation data, or the target dataset.

4.2 Parallel Texts

Scalability is a problem faced by current supervised WSD systems, as they usually rely on manually annotated data for training. To tackle this problem, in one of our recent work (Ng et al., 2003), we had gathered training data from parallel texts and obtained encouraging results in our

evaluation on the nouns of SENSEVAL-2 English lexical sample task (Kilgarriff, 2001). In another recent evaluation on the nouns of SENSEVAL-2 English all-words task (Chan and Ng, 2005a), promising results were also achieved using examples gathered from parallel texts. Due to the potential of parallel texts in addressing the issue of scalability, we also drew training data for our earlier sense priors estimation experiments (Chan and Ng, 2005b) from parallel texts. In addition, our parallel texts training data represents a natural domain difference with the test data of SENSEVAL-2 English lexical sample task, of which 91% is drawn from the British National Corpus (BNC).

As part of our experiments, we followed the experimental setup of our earlier work (Chan and Ng, 2005b), using the same 6 English-Chinese parallel corpora (*Hong Kong Hansards*, *Hong Kong News*, *Hong Kong Laws*, *Sinorama*, *Xinhua News*, and *English translation of Chinese Treebank*), available from Linguistic Data Consortium. To gather training examples from these parallel texts, we used the approach we described in (Ng et al., 2003) and (Chan and Ng, 2005b). We then evaluated our estimation of sense priors on the nouns of SENSEVAL-2 English lexical sample task, similar to the evaluation we conducted in (Chan and Ng, 2005b). Since the test data for the nouns of SENSEVAL-3 English lexical sample task (Mihalcea et al., 2004) were also drawn from BNC and represented a difference in domain from the parallel texts we used, we also expanded our evaluation to these SENSEVAL-3 nouns.

4.3 Choice of Words

Research by (McCarthy et al., 2004) highlighted that the sense priors of a word in a corpus depend on the domain from which the corpus is drawn. A change of predominant sense is often indicative of a change in domain, as different corpora drawn from different domains usually give different predominant senses. For example, the predominant sense of the noun *interest* in the BC part of the DSO corpus has the meaning “a sense of concern with and curiosity about someone or something”. In the WSJ part of the DSO corpus, the noun *interest* has a different predominant sense with the meaning “a fixed charge for borrowing money”, reflecting the business and finance focus of the WSJ corpus.

Estimation of sense priors is important when

there is a significant change in sense priors between the training and target dataset, such as when there is a change in domain between the datasets. Hence, in our experiments involving the DSO corpus, we focused on the set of nouns and verbs which had different predominant senses between the BC and WSJ parts of the corpus. This gave us a set of 37 nouns and 28 verbs. For experiments involving the nouns of SENSEVAL-2 and SENSEVAL-3 English lexical sample task, we used the approach we described in (Chan and Ng, 2005b) of sampling training examples from the parallel texts using the natural (empirical) distribution of examples in the parallel texts. Then, we focused on the set of nouns having different predominant senses between the examples gathered from parallel texts and the evaluation data for the two SENSEVAL tasks. This gave a set of 6 nouns for SENSEVAL-2 and 9 nouns for SENSEVAL-3. For each noun, we gathered a maximum of 500 parallel text examples as training data, similar to what we had done in (Chan and Ng, 2005b).

5 Experimental Results

Similar to our previous work (Chan and Ng, 2005b), we used the supervised WSD approach described in (Lee and Ng, 2002) for our experiments, using the naive Bayes algorithm as our classifier. Knowledge sources used include parts-of-speech, surrounding words, and local collocations. This approach achieves state-of-the-art accuracy. All accuracies reported in our experiments are micro-averages over all test examples.

In (Chan and Ng, 2005b), we used a multiclass naive Bayes classifier (denoted by *NB*) for each word. Following this approach, we noted the WSD accuracies achieved without any adjustment, in the column *L* under *NB* in Table 1. The predictions $\hat{p}_L(\omega_i | \mathbf{x}_k)$ of these naive Bayes classifiers are then used in Equation (2) and (3) to estimate the sense priors $\hat{p}(\omega_i)$, before being adjusted by these estimated sense priors based on Equation (4). The resulting WSD accuracies after adjustment are listed in the column *EM_{NB}* in Table 1, representing the WSD accuracies achievable by following the approach we described in (Chan and Ng, 2005b).

Next, we used the one-against-all approach to reduce each multiclass problem into a set of binary class problems. We trained a naive Bayes classifier for each binary problem and calibrated the probabilities from these binary classifiers. The WSD

Classifier	NB			NBcal		
	L	EM_{NB}	EM_{LogR}	L	EM_{NBcal}	EM_{LogR}
DSO nouns	44.5	46.1	46.6	45.8	47.0	51.1
DSO verbs	46.7	48.3	48.7	46.9	49.5	50.8
SE2 nouns	61.7	62.4	63.0	62.3	63.2	63.5
SE3 nouns	53.9	54.9	55.7	55.4	58.8	58.4

Table 1: Micro-averaged WSD accuracies using the various methods. The different naive Bayes classifiers are: multiclass naive Bayes (NB) and naive Bayes with calibrated probabilities (NBcal).

Dataset	True - L	$EM_{NBcal} - L$	$EM_{LogR} - L$
DSO nouns	11.6	1.2 (10.3%)	5.3 (45.7%)
DSO verbs	10.3	2.6 (25.2%)	3.9 (37.9%)
SE2 nouns	3.0	0.9 (30.0%)	1.2 (40.0%)
SE3 nouns	3.7	3.4 (91.9%)	3.0 (81.1%)

Table 2: Relative accuracy improvement based on calibrated probabilities.

Dataset	EM_{NB}	EM_{NBcal}	EM_{LogR}
DSO nouns	0.621	0.586	0.293
DSO verbs	0.651	0.602	0.307
SE2 nouns	0.371	0.307	0.214
SE3 nouns	0.693	0.632	0.408

Table 3: KL divergence between the true and estimated sense distributions.

accuracies of these calibrated naive Bayes classifiers (denoted by *NBcal*) are given in the column *L* under *NBcal*.¹ The predictions of these classifiers are then used to estimate the sense priors $\hat{p}(\omega_i)$, before being adjusted by these estimates based on Equation (4). The resulting WSD accuracies after adjustment are listed in column EM_{NBcal} in Table 1.

The results show that calibrating the probabilities improves WSD accuracy. In particular, EM_{NBcal} achieves the highest accuracy among the methods described so far. To provide a basis for comparison, we also adjusted the calibrated probabilities by the *true* sense priors $p(\omega_i)$ of the test data. The increase in WSD accuracy thus obtained is given in the column *True - L* in Table 2. Note that this represents the maximum possible increase in accuracy achievable provided we know these *true* sense priors $p(\omega_i)$. In the column $EM_{NBcal} - L$ in Table 2, we list the increase in WSD accuracy when adjusted by the sense priors $\hat{p}(\omega_i)$ which were *automatically* estimated using the EM procedure. The relative improvements obtained with using $\hat{p}(\omega_i)$ (compared against using $p(\omega_i)$) are given as percentages in brackets. As an example, according to Table 1 for the DSO verbs, EM_{NBcal} gives an improvement of $49.5\% - 46.9\% = 2.6\%$ in WSD accuracy, and the relative improvement compared to using the *true* sense priors is $2.6/10.3 = 25.2\%$, as shown in Table 2.

6 Discussion

The experimental results show that the sense priors estimated using the calibrated probabilities of naive Bayes are effective in increasing the WSD accuracy. However, using a learning algorithm which already gives well calibrated posterior probabilities may be more effective in estimating the sense priors. One possible algorithm is logistic regression, which directly optimizes for getting approximations of the posterior probabilities. Hence, its probability estimates are already well calibrated (Zhang and Yang, 2004; Niculescu-Mizil and Caruana, 2005).

In the rest of this section, we first conduct experiments to estimate sense priors using the predictions of logistic regression. Then, we perform significance tests to compare the various methods.

6.1 Using Logistic Regression

We trained logistic regression classifiers and evaluated them on the 4 datasets. However, the WSD accuracies of these unadjusted logistic regression classifiers are on average about 4% lower than those of the unadjusted naive Bayes classifiers. One possible reason is that being a discriminative learner, logistic regression requires more training examples for its performance to catch up to, and possibly overtake the generative naive Bayes learner (Ng and Jordan, 2001).

Although the accuracy of logistic regression as a basic classifier is lower than that of naive Bayes, its predictions may still be suitable for estimating

¹Though not shown, we also calculated the accuracies of these binary classifiers without calibration, and found them to be similar to the accuracies of the multiclass naive Bayes shown in the column *L* under *NB* in Table 1.

Method comparison	DSO nouns	DSO verbs	SE2 nouns	SE3 nouns
NB-EM _{LogR} vs. NB-EM _{NB}	≫	≫	≫	≫
NBcal-EM _{NBcal} vs. NB-EM _{NB}	~	≫	>	≫
NBcal-EM _{NBcal} vs. NB-EM _{LogR}	~	≫	~	≫
NBcal-EM _{LogR} vs. NB-EM _{NB}	≫	≫	≫	≫
NBcal-EM _{LogR} vs. NB-EM _{LogR}	≫	≫	~	≫
NBcal-EM _{LogR} vs. NBcal-EM _{NBcal}	≫	≫	~	~

Table 4: Paired t-tests between the various methods for the 4 datasets.

sense priors. To gauge how well the sense priors are estimated, we measure the KL divergence between the true sense priors and the sense priors estimated by using the predictions of (uncalibrated) multiclass naive Bayes, calibrated naive Bayes, and logistic regression. These results are shown in Table 3 and the column EM_{LogR} shows that using the predictions of logistic regression to estimate sense priors consistently gives the lowest KL divergence.

Results of the KL divergence test motivate us to use sense priors estimated by logistic regression on the predictions of the naive Bayes classifiers. To elaborate, we first use the probability estimates $\hat{p}_L(\omega_i|\mathbf{x}_k)$ of logistic regression in Equations (2) and (3) to estimate the sense priors $\hat{p}(\omega_i)$. These estimates $\hat{p}(\omega_i)$ and the predictions $\hat{p}_L(\omega_i|\mathbf{x}_k)$ of the calibrated naive Bayes classifier are then used in Equation (4) to obtain the adjusted predictions. The resulting WSD accuracy is shown in the column EM_{LogR} under $NBcal$ in Table 1. Corresponding results when the predictions $\hat{p}_L(\omega_i|\mathbf{x}_k)$ of the multiclass naive Bayes is used in Equation (4), are given in the column EM_{LogR} under NB . The relative improvements against using the true sense priors, based on the calibrated probabilities, are given in the column $EM_{LogR} - L$ in Table 2. The results show that the sense priors provided by logistic regression are in general effective in further improving the results. In the case of DSO nouns, this improvement is especially significant.

6.2 Significance Test

Paired t-tests were conducted to see if one method is significantly better than another. The t statistic of the difference between each test instance pair is computed, giving rise to a p value. The results of significance tests for the various methods on the 4 datasets are given in Table 4, where the symbols “~”, “>”, and “≫” correspond to p-value > 0.05 , $(0.01, 0.05]$, and ≤ 0.01 respectively.

The methods in Table 4 are represented in the form $a1$ - $a2$, where $a1$ denotes adjusting the pre-

dictions of which classifier, and $a2$ denotes how the sense priors are estimated. As an example, NBcal-EM_{LogR} specifies that the sense priors estimated by logistic regression is used to adjust the predictions of the calibrated naive Bayes classifier, and corresponds to accuracies in column EM_{LogR} under $NBcal$ in Table 1. Based on the significance tests, the adjusted accuracies of EM_{NB} and EM_{NBcal} in Table 1 are significantly better than their respective unadjusted L accuracies, indicating that estimating the sense priors of a new domain via the EM approach presented in this paper significantly improves WSD accuracy compared to just using the sense priors from the old domain.

NB-EM_{NB} represents our earlier approach in (Chan and Ng, 2005b). The significance tests show that our current approach of using calibrated naive Bayes probabilities to estimate sense priors, and then adjusting the calibrated probabilities by these estimates (NBcal-EM_{NBcal}) performs significantly better than NB-EM_{NB} (refer to row 2 of Table 4). For DSO nouns, though the results are similar, the p value is a relatively low 0.06.

Using sense priors estimated by logistic regression further improves performance. For example, row 1 of Table 4 shows that adjusting the predictions of multiclass naive Bayes classifiers by sense priors estimated by logistic regression (NB-EM_{LogR}) performs significantly better than using sense priors estimated by multiclass naive Bayes (NB-EM_{NB}). Finally, using sense priors estimated by logistic regression to adjust the predictions of calibrated naive Bayes (NBcal-EM_{LogR}) in general performs significantly better than most other methods, achieving the best overall performance.

In addition, we implemented the unsupervised method of (McCarthy et al., 2004), which calculates a prevalence score for each sense of a word to predict the predominant sense. As in our earlier work (Chan and Ng, 2005b), we normalized the prevalence score of each sense to obtain estimated sense priors for each word, which we then used

to adjust the predictions of our naive Bayes classifiers. We found that the WSD accuracies obtained with the method of (McCarthy et al., 2004) are on average 1.9% lower than our NBcal-EM_{LogR} method, and the difference is statistically significant.

7 Conclusion

Differences in sense priors between training and target domain datasets will result in a loss of WSD accuracy. In this paper, we show that using well calibrated probabilities to estimate sense priors is important. By calibrating the probabilities of the naive Bayes algorithm, and using the probabilities given by logistic regression (which is already well calibrated), we achieved significant improvements in WSD accuracy over previous approaches.

References

- Eneko Agirre and David Martinez. 2004. Unsupervised WSD based on automatically retrieved examples: The importance of bias. In *Proc. of EMNLP04*.
- Miriam Ayer, H. D. Brunk, G. M. Ewing, W. T. Reid, and Edward Silverman. 1955. An empirical distribution function for sampling with incomplete information. *Annals of Mathematical Statistics*, 26(4).
- Yee Seng Chan and Hwee Tou Ng. 2005a. Scaling up word sense disambiguation via parallel texts. In *Proc. of AAAI05*.
- Yee Seng Chan and Hwee Tou Ng. 2005b. Word sense disambiguation with distribution estimation. In *Proc. of IJCAI05*.
- Pedro Domingos and Michael Pazzani. 1996. Beyond independence: Conditions for the optimality of the simple Bayesian classifier. In *Proc. of ICML-1996*.
- Gerard Escudero, Lluís Marquez, and German Rigau. 2000. An empirical study of the domain dependence of supervised word sense disambiguation systems. In *Proc. of EMNLP/VLC00*.
- Adam Kilgarriff. 2001. English lexical sample task description. In *Proc. of SENSEVAL-2*.
- Yoong Keok Lee and Hwee Tou Ng. 2002. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proc. of EMNLP02*.
- Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant word senses in untagged text. In *Proc. of ACL04*.
- Rada Mihalcea, Timothy Chklovski, and Adam Kilgarriff. 2004. The senseval-3 english lexical sample task. In *Proc. of SENSEVAL-3*.
- George A. Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G. Thomas. 1994. Using a semantic concordance for sense identification. In *Proc. of ARPA Human Language Technology Workshop*.
- Andrew Y. Ng and Michael I. Jordan. 2001. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Proc. of NIPS14*.
- Hwee Tou Ng and Hian Beng Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proc. of ACL96*.
- Hwee Tou Ng, Bin Wang, and Yee Seng Chan. 2003. Exploiting parallel texts for word sense disambiguation: An empirical study. In *Proc. of ACL03*.
- Alexandru Niculescu-Mizil and Rich Caruana. 2005. Predicting good probabilities with supervised learning. In *Proc. of ICML05*.
- Tim Robertson, F. T. Wright, and R. L. Dykstra. 1988. Chapter 1. Isotonic Regression. In *Order Restricted Statistical Inference*. John Wiley & Sons.
- Marco Saerens, Patrice Latinne, and Christine Decaestecker. 2002. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Computation*, 14(1).
- Slobodan Vucetic and Zoran Obradovic. 2001. Classification on data with biased class distribution. In *Proc. of ECML01*.
- Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proc. of KDD02*.
- Jian Zhang and Yiming Yang. 2004. Probabilistic score estimation with piecewise logistic regression. In *Proc. of ICML04*.