Active Learning for Word Sense Disambiguation with Methods for Addressing the Class Imbalance Problem

Jingbo Zhu
University of Southern California
Information Sciences Institute
Northeastern University, P.R.China
Natural Language Processing Lab
Zhujingbo@mail.neu.edu.cn

Eduard Hovy
University of Southern California
Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292-6695
hovy@isi.edu

Abstract

In this paper, we analyze the effect of resampling techniques, including under-sampling and over-sampling used in active learning for word sense disambiguation (WSD). Experimental results show that under-sampling causes negative effects on active learning, but over-sampling is a relatively good choice. To alleviate the within-class imbalance problem of over-sampling, we propose a bootstrap-based over-sampling (BootOS) method that works better than ordinary over-sampling in active learning for WSD. Finally, we investigate when to stop active learning, and adopt two strategies, max-confidence and min-error, as stopping conditions for active learning. According to experimental results, we suggest a prediction solution by considering max-confidence as the upper bound and min-error as the lower bound for stopping conditions.

1 Introduction

Word sense ambiguity is a major obstacle to accurate information extraction, summarization, and machine translation (Ide and Veronis, 1998). In recent years, a variety of techniques for machine learning algorithms have demonstrated remarkable performance for automated word sense disambiguation (WSD) (Chan and Ng, 2006; Dagan et. al., 2006; Xue *et. al.*, 2006; Kohomban and Lee. 2005; Dang and Palmer, 2005), when enough labeled training data is available. However, creating a large sense-tagged corpus is very expensive and time-consuming, because these data have to be annotated by human experts.

Among the techniques to solve the knowledge bottleneck problem, active learning is a promising way (Lewis and Gale, 1994; McCallum and Nigram, 1998). The purpose of active learning is to minimize the amount of human labeling effort by having the system automatically select for human annotation the most informative unannotated case.

In real-world data, the distribution of the senses of a word is often very skewed. Some studies reported that simply selecting the predominant sense provides superior performance, when a highly skewed sense distribution and insufficient context exist (Hoste *et al.*, 2001; McCarthy *et. al.*, 2004). The data set is *imbalanced* when at least one of the senses is heavily underrepresented compared to the other senses. In general, a WSD classifier is designed to optimize overall accuracy without taking into account the class imbalance distribution in a real-world data set. The result is that the classifier induced from imbalanced data tends to over-fit the predominant class and to ignore small classes (Japkowicz and Stephen, 2002). Recently, much work has been done in addressing the class imbalance problem, reporting that resampling methods such as *over-sampling* and *under-sampling* are useful in supervised learning with imbalanced data sets to induce more effective classifiers (Estabrooks *et al.*, 2004; Zhou and Liu, 2006).

In general framework of active learning, the learner (i.e. supervised classifier) is formed by using supervised learning algorithms. To date, however, no-one has studied the effects of over-sampling and under-sampling on active learning

Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 783–790, Prague, June 2007.

methods. In this paper, we study active learning with resampling methods addressing the class imbalance problem for WSD. It is noteworthy that neither of these techniques need modify the architecture or learning algorithm, making them very easy to use and extend to other domains.

Another problem in active learning is knowing when to stop the process. We address this problem in this paper, and discuss how to form the final classifier for use. This is a problem of estimation of classifier effectiveness (Lewis and Gale, 1994). Because it is difficult to know when the classifier reaches maximum effectiveness, previous work used a simple stopping condition when the training set reaches desirable size. However, in fact it is almost impossible to predefine an appropriate size of desirable training data for inducing the most effective classifier. To solve the problem, we consider the problem of estimation of classifier effectiveness as a second task of estimating classifier confidence. This paper adopts two strategies: *max-confidence* and *min-error*, and suggests a prediction solution by considering max-confidence as the upper bound and min-error as the lower bound for the stopping conditions.

2 Related Work

The ability of the active learner can be referred to as *selective sampling*, of which two major schemes exist: *uncertainty sampling* and *committee-based sampling*. The former method, for example proposed by Lewis and Gale (1994), is to use only one classifier to identify unlabeled examples on which the classifier is least confident. The latter method (McCallum and Nigam, 1998) generates a committee of classifiers (always more than two classifiers) and selects the next unlabeled example by the principle of maximal disagreement among these classifiers. With selective sampling, the size of the training data can be significantly reduced for text classification (Lewis and Gale, 1994; McCallum and Nigam, 1998), and word sense disambiguation (Chen, *et al.* 2006).

A method similar to committee-based sampling is *co-testing* proposed by Muslea et al. (2000), which trains two learners individually on two compatible and uncorrelated views that should be able to reach the same classification accuracy. In practice, however, these conditions of view selection are difficult to meet in real-world word sense disambiguation tasks.

Recently, much work has been done on the class imbalance problem. The well-known approach is *resampling*, in which some training material is duplicated. Two types of popular resampling methods exist for addressing the class imbalance problem: *over-sampling* and *under-sampling*. The basic idea of resampling methods is to change the training data distribution and make the data more balanced. It works ok in supervised learning, but has not been tested in active learning. Previous work reports that cost-sensitive learning is a good solution to the class imbalance problem (Weiss, 2004). In practice, for WSD, the costs of various senses of a disambiguated word are unequal and unknown, and they are difficult to evaluate in the process of learning.

In recent years, there have been attempts to apply active learning for word sense disambiguation (Chen *et al.*, 2006). However, to our best knowledge, there has been no such attempt to consider the class imbalance problem in the process of active learning for WSD tasks.

3 Resampling Methods

3.1 Under-sampling

Under-sampling is a popular method in addressing the class imbalance problem by changing the training data distribution by removing some exemplars of the majority class at random. Some previous work reported that under-sampling is effective in learning on large imbalanced data sets (Japkowicz and Stephen, 2002). However, as under-sampling removes some potentially useful training samples, it could cause negative effects on the classifier performance.

One-sided sampling is a method similar to under-sampling, in which redundant and borderline training examples are identified and removed from training data (Kubat and Matwin, 1997). Kuban and Matwin reported that one-sided sampling is effective in learning with two-class large imbalanced data sets. However, the relative computational cost of one-sided sampling in active learning is very high, because sampling computations must be implemented for each learning iteration. Our primitive experimental results show that, in the multi-class problem of WSD, one-sided sampling degrades the performance of active learning. And due to the high computation complexity of one-sided sampling, we use random under-sampling in our comparison experiments instead.

To control the degree of change of the training data distribution, the ratio of examples from the majority and the minority class after removal from the majority class is called the *removal rate* (Jo and Japkowicz, 2004). If the removal rate is 1.0, then under-sampling methods build data sets with complete class balance. However, it was reported previously that perfect balance is not always the optimal rate (Estabrooks *et al.*, 2004). In our comparison experiments, we set the removal rate for under-sampling to 0.8, since some cases have 0.8 as the optimal rate reported in (Estabrooks *et al.*, 2004).

3.2 Over-sampling

Over-sampling is also a popular method in addressing the class imbalance problem by resampling the small class until it contains as many examples as the large one. In contrast to under-sampling, over-sampling is the process of adding examples to the minority class, and is accomplished by random sampling and duplication. Because the process of over-sampling involves making exact copies of examples, it usually increases the training cost and may lead to overfitting. There is a recent variant of over-sampling named SMOTE (Chawla *et al.*, 2002) which is a synthetic minority over-sampling technique. The authors reported that a combination of SMOTE and under-sampling can achieve better classifier performance in ROC space than only under-sampling the majority class.

In our comparison experiments, we use over-sampling, measured by a resampling rate called the *addition rate* (Jo and Japkowicz, 2004) that indicates the number of examples that should be added into the minority class. The addition rate for over-sampling is also set to 0.8 in our experiments.

3.3 Bootstrap-based Over-sampling

While over-sampling decreases the between-class imbalance, it increases the within-class imbalance (Jo and Japkowicz, 2004) because of the increase of exact copies of examples at random. To alleviate this within-class imbalance problem, we propose a *bootstrap-based over-sampling* method (BootOS) that uses a bootstrap resampling technique in the process of over-sampling. Bootstrapping, explained below, is a resampling technique similar to *jackknifing*.

There are two reasons for choosing a bootstrap method as resampling technique in the process of over-sampling. First, using a bootstrap set can avoid exactly copying samples in the minority class. Second, the bootstrap method may give a smoothing of the distribution of the training samples (Hamamoto *et al.*, 1997), which can alleviate the within-class imbalance problem cased by over-sampling.

To generate the bootstrap set, we use a well-known bootstrap technique proposed by Hamamoto *et al.* (1997) that does not select samples randomly, allowing all samples in the minority class(es) an equal chance to be selected.

Algorithm *BootOS*(X, N, r, k)
Input: Minority class sample set X={x_1, x_2, …, x_n} of size n; Difference in number of examples between the majority and the minority class = N; Addition rate = r (< 1.0); Number of nearest neighbors = k.
Output: bootstrap sample set X_B of size N*r =X ∪ (x_{B1}, x_{B2}, …, $x_{B(N*r)}$).

For i = 1 **To** N*r
  If i == n then (*all samples in minority class sample set have been used*)
    j = 1; //the first sample is selected again
  Else
    j = i; // the i-th sample is selected
  Endif
  Select j-th sample x_j (also as $x_{j,0}$) from X
  Find the k nearest neighbor samples $x_{j,1}$, $x_{j,2}$, …, $x_{j,k}$ using similarity functions.
  Compute a bootstrap sample x_{Bi}:

$$x_{Bi} = \frac{1}{k+1}\sum_{l=0}^{k} x_{j,l}$$

Endfor
return

Figure 1. The BootOS algorithm

4 Active Learning with Resampling

In this work, we are interested in selective sampling for pool-based active learning, and focus on uncertainty sampling (Lewis and Gale, 1994). The key point is how to measure the uncertainty of an unlabeled exemplar, and select a new exemplar with maximum uncertainty to augment the training data. The maximum uncertainty implies that the current classifier has the least confidence in its classification of this exemplar. The well-known *entropy* is a good uncertainty measurement widely used in active learning (zhang and Chen, 2002; Chen *et al.*, 2006):

$$U(i) = H(P_i) = -\sum_{j=1}^{n_i} p(s_j \mid w_i) \log p(s_j \mid w_i) \quad (1)$$

where U is the uncertainty measurement function H represents the entropy function. In the WSD task, $p(s_j|w_i)$ is the predicted probability of sense s_j outputted by the current classifier, when given a sample i containing a disambiguated word w_i.

Algorithm *Active-Learning-with-Resampling*(*L,U,m*)
Input: Let L be initial small training data set; U the pool of unlabeled exemplars
Output: labeled training data set L

1. Resample L to generate new training data set L^* using resampling techniques such as under-sampling, over-sampling or BootOS, and then use L^* to train the initial classifier
2. **Loop** while adding new instances into L
   a. use the current classifier to probabilistically label all unlabeled exemplars in U
   b. Based on active learning rules, present *m* top-ranked exemplars to *oracle* for labeling
   c. Augment L with the *m* new exemplars, and remove them from U
   d. Resample L to generate new training data set L^* using resampling techniques such as under-sampling, over-sampling, or BootOS, and use L^* to retrain the current classifier
3. **Until** the predefined stopping condition is met.
4. **return**

Figure 2. Active learning with resampling

In step 1 and 2(d) in Fig. 2, if we do not generate L*, and L is used directly to train the current classifier, we call it *ordinary active learning*. In the process of active learning, we used the entropy-based uncertainty measurement for all active learning frameworks in our comparison experiments. Actually our active learning with resampling is a heterogeneous approach in which the classifier used to select new instances is different from the resulting classifier (Lewis and Catlett, 1994).

We utilize a maximum entropy (ME) model (Berger *et al.*, 1996) to design the basic classifier used in active learning for WSD. The advantage of the ME model is the ability to freely incorporate features from diverse sources into a single, well-grounded statistical model. A publicly available ME toolkit (Zhang *et. al.*, 2004) was used in our experiments. In order to extract the linguistic features necessary for the ME model, all sentences containing the target word were automatically part-of-speech (POS) tagged using the Brill POS tagger (Brill, 1992). Three knowledge sources were used to capture contextual information: unordered single words in topical context, POS of neighboring words with position information, and local collocations. These are same as three of the four knowledge sources used in (Lee and Ng, 2002). Their fourth knowledge source (named syntactic relations) was not used in our work.

5 Stopping Conditions

In active learning algorithm, defining the stopping condition for active learning is a critical problem, because it is almost impossible for the human annotator to label all unlabeled samples. This is a problem of estimation of classifier effectiveness (Lewis and Gale 1994). In fact, it is difficult to know when the classifier reaches maximum effectiveness. In previous work some researchers used a simple stopping condition when the training set reached a predefined desired size. It is almost impossible to predefine an appropriate size of desirable training data for inducing the most effective classifier.

To solve the problem, we consider the problem of estimating classifier effectiveness as the problem of confidence estimation of classifier on the remaining unlabeled samples. Concretely, if we find that the current classifier already has acceptably strong confidence on its classification results for all remained unlabeled data, we assume the current training data is sufficient to train the classifier with maximum effectiveness. In other words, if a classifier induced from the current training data has strong classification confidence on an unlabeled example, we could consider it as a redundant example.

Based on above analyses, we adopt here two stopping conditions for active learning:

- ***Max-confidence***: This strategy is based on uncertainty measurement, considering whether the entropy of each selected unlabeled example is less than a very small predefined threshold close to zero, such as 0.001.
- ***Min-error***: This strategy is based on feedback from the *oracle* when the active learner asks for true labels for selected unlabeled examples, considering whether the current trained classifier could correctly predict the labels or the accuracy performance of predictions on

selected unlabeled examples is already larger than a predefined accuacy threshold.

Once *max-confidence* and *min-error* conditions are met, the current classifier is assumed to have strong enough confidence on the classification results of all remained unlabeled data.

6 Evaluation

6.1 Data

The data used for our comparison experiments were developed as part of the OntoNotes project (Hovy *et al.*, 2006), which uses the WSJ part of the Penn Treebank (Marcus *et al.*, 1993). The senses of noun words occurring in OntoNotes are linked to the Omega ontology. In OntoNotes, at least two humans manually annotate the coarse-grained senses of selected nouns and verbs in their natural sentence context. To date, OntoNotes has annotated several tens of thousands of examples, covering several hundred nouns and verbs, with an inter-annotator agreement rate of at least 90%.

Those 38 random chosen ambiguous nouns used in all following experiments are shown in Table 1. It is apparent that the sense distributions of most nouns are very skewed (frequencies shown in the table, separated by /).

Words	sense distribution	words	sense distribution
Rate	1025/182	president	936/157/17
People	815/67/7/5	part	456/102/75/16
Point	471/88/37/19/9/6	director	517/23
Revenue	517/23	bill	348/130/40
Future	413/82/23	order	354/61/54/6/6
Plant	376/51	board	369/15
Today	238/149	policy	308/74
Capital	325/21/8	term	147/137/52/13
management	210/130	move	302/13/5
Position	97/75/67/61/10/7	amount	236/57/16
Home	267/17/16	power	154/134/15
Leader	244/38	return	191/35/29/12/9
administration	266/11	payment	201/69
Account	233/18/13	control	90/66/64/21/12/5
Lot	221/20	activity	218/23
Drug	160/74	building	177/48/5
Estate	214/11	house	112/71/25
development	165/46/6	network	127/53/29
Strategy	198/11	place	69/63/50/18/5

Table 1. Data set used in experiments

6.2 Results

In the following active learning comparison experiments, we tested with five resampling methods including *random sampling* (Random), *uncertainty sampling* (Ordinary), *under-sampling*, *over-sampling*, and *BootOS*. The 1-NN technique was used for bootstrap-based resampling of BootOS in our experiments. A 5 by 5-fold cross-validation was performed on each noun's data.

We used 20% randomly chosen data for held-out evaluation and the other 80% as the pool of unlabeled data for each round of the active learning. For all words, we started with a randomly chosen initial training set of 10 examples, and we made 10 queries after each learning iteration.

In the evaluation, average *accuracy* and *recall* are used as measures of performances for each active learning method. Note that the *macro-average* way is adopted for recall evaluation in each noun WSD task. The accuracy measure indicates the percentage of testing instances correctly identified by the system. The macro-average recall measure indicates how well the system performs on each sense.

Experiment 1: Performance comparison experiments on active learning

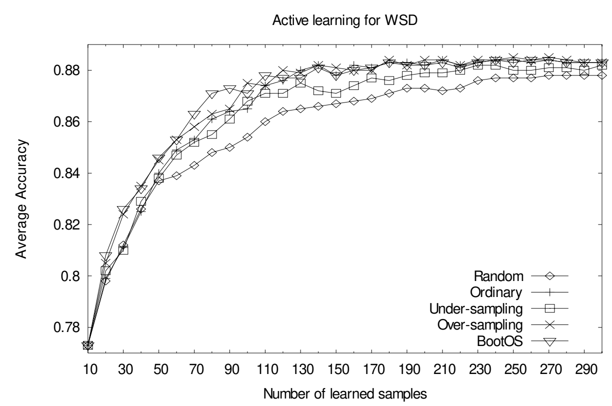

Figure 3. Average accuracy performance comparison experiments

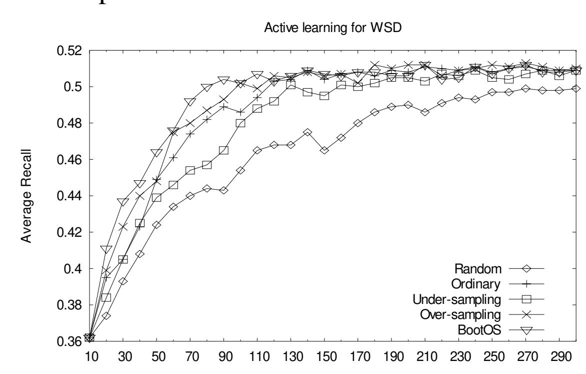

Figure 4. Average recall performance comparison experiments

As shown in Fig. 3 and Fig. 4, when the number of learned samples for each noun is smaller than 120, the BootOS has the best performance, followed by over-sampling and ordinary method. As the number of learned samples increases, ordinary, over-sampling and BootOS have similar performances on accuracy and recall. Our experiments also exhibit that random sampling method is the worst on both accuracy and recall.

Previous work (Estabrooks *et al.*, 2004) reported that under-sampling of the majority class (predominant sense) has been proposed as a good means of increasing the sensitivity of a classifier to the minority class (infrequent sense). However, in our active learning experiments, under-sampling is apparently worse than ordinary, over-sampling and our BootOS. The reason is that in highly imbalanced data, too many useful training samples of majority class are discarded in under-sampling, causing the performance of active learning to degrade.

Experiment 2: Effectiveness of learning instances for infrequent senses

It is important to enrich the corpora by learning more instances for infrequent senses using active learning with less human labeling. This procedure not only makes the corpora 'richer', but also alleviates the domain dependence problem faced by corpus-based supervised approaches to WSD.

The objective of this experiment is to evaluate the performance of active learning in learning samples of infrequent senses from an unlabeled corpus. Due to highly skewed word sense distributions in our data set, we consider all senses other than the predominant sense as infrequent senses in this experiment.

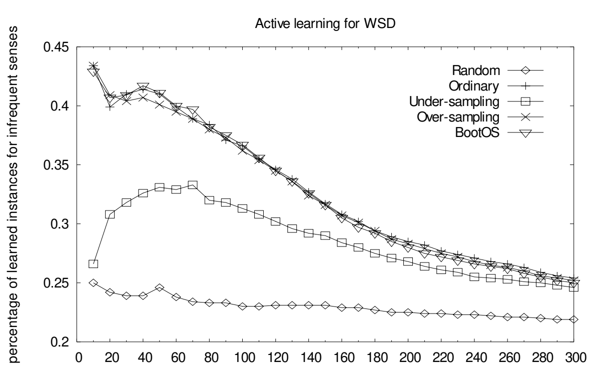

Figure 5. Comparison experiments on learning instances for infrequent senses

Fig. 5 shows that random sampling is the worst in active learning for infrequent senses. The reason is very obvious: the sense distribution of the learned sample set by random sampling is almost identical to that of the original data set.

Under-sampling is apparently worse than ordinary active learning, over-sampling and BootOS methods. When the number of learned samples for each noun is smaller than 80, BootOS achieves slight better performance than ordinary active learning and over-sampling.

When the number of learned samples is larger than 80 and smaller than 160, these three methods exhibit similar performance. As the number of iterations increases, ordinary active learning is slightly better than over-sampling and BootOS. In fact, after the 16th iteration (10 samples chosen in each iteration), results indicate that most instances for infrequent senses have been learned.

Experiment 3: Effectiveness of Stopping Conditions for active learning

To evaluate the effectiveness of two strategies *max-confidence* and *min-error* as stopping conditions of active learning, we first construct an ideal stopping condition when the classifier could reach the highest accuracy performance at the first time in the procedure of active learning. When the ideal stopping condition is met, it means that the current classifier has reached maximum effectiveness. In practice, it is impossible to exactly know when the ideal stopping condition is met before all unlabeled data are labeled by a human annotator. We only use this ideal method in our comparison experiments to analyze the effectiveness of our two proposed stopping conditions.

For general purpose, we focus on the ordinary active learning to design the basic system, and to evaluate the effectiveness of three stop conditions. In the following experiments, the entropy threshold used in *max-confidence* strategy is set to 0.001, and the accuracy threshold used in *min-error strategy* is set to 0.9.

In Table 2, the column "*Size*" stands for the size of unlabeled data set of corresponding noun word used in active learning. There are two columns for each stopping condition: the left column "*num*" presents number of learned instances and the right column *"%"* presents its percentage over all data when the corresponding stopping condition is met.

Words	Size	Ideal		Max-confidence		Min-error	
		num	%	num	%	num	%
Rate	966	200	.23	410	.41	290	.29
People	715	140	.20	290	.41	200	.28
Point	504	90	.18	220	.44	120	.24
Revenue	432	70	.16	110	.25	80	.19
Future	414	120	.29	140	.34	60	.14
Plant	342	210	.61	180	.53	110	.32
Today	382	250	.65	240	.63	230	.60
Capital	283	70	.25	180	.64	90	.32
Management	272	200	.74	210	.77	210	.77
Position	254	210	.83	230	.91	220	.87
Home	240	60	.25	160	.67	60	.25
Leader	226	60	.27	120	.53	70	.31
administration	222	30	.14	90	.41	50	.23
Account	211	50	.24	130	.62	70	.33
Lot	185	30	.16	60	.32	40	.22
Drug	187	130	.70	140	.75	120	.64
Estate	180	20	.11	50	.28	30	.17
Development	174	40	.23	150	.86	80	.46
Strategy	167	10	.06	100	.60	10	.06
President	888	120	.14	220	.25	120	.14
Part	519	110	.21	240	.46	130	.25
Director	432	110	.25	130	.30	90	.21
Bill	414	120	.29	280	.68	150	.36
Order	385	130	.34	220	.57	140	.36
Board	307	40	.13	190	.62	40	.13
Policy	306	90	.29	200	.65	150	.49
Term	279	120	.43	190	.68	130	.47
Move	256	50	.20	140	.55	50	.20
Amount	247	210	.85	200	.81	140	.57
Power	242	190	.78	190	.78	190	.78
Return	221	90	.41	160	.72	100	.45
Payment	216	120	.56	160	.74	150	.69
Control	206	160	.78	200	.97	200	.97
Activity	193	30	.16	130	.67	70	.36
Building	184	90	.49	130	.71	110	.60
House	166	100	.60	150	.90	110	.66
Network	167	110	.66	130	.78	100	.60
Place	164	120	.73	150	.91	120	.73

Table 2 Effectiveness of three stopping conditions

As shown in Table 2, the min-error strategy based on feedback of human annotator is very close to the ideal method. Therefore, when comparing to ideal stopping condition, min-error strategy is a good choice as stopping condition for active learning. It is important to note that the min-error method does not need more additional computational costs, it only depends upon the feedback of human annotator when labeling the chosen unlabeled samples.

From experimental results, we can see that max-confidence strategy is worse than min-error method. However, we believe that the entropy of each unlabeled sample is a good signal to stop active learning. So we suggest that there may be a good prediction solution in which the min-error strategy is used as the lower-bound of stopping condition, and max-confidence strategy as the upper-bound of stopping condition for active learning.

7 Discussion

As discussed above, finding more instances for infrequent senses at the earlier stages of active learning is very significant in making the corpus richer, meaning less effort for human labeling. In practice, another way to learn more instances for infrequent senses is to first build a training data set by active learning or by human efforts, and then build a supervised classifier to find more instances for infrequent sense. However, it is interesting to know how much initial training data is enough for this task, and how much human labeling efforts could be saved.

From experimental results, we found that among these chosen unlabeled instances by active learner, some instances are informative samples helpful for improving classification performance, and other instances are borderline samples which are unreliable because even a small amount of noise can lead the sample to the wrong side of the decision boundary. The removal of these borderline samples might improve the performance of active learning.

The proposed prediction solution based on max-confidence and min-error strategies is a coarse framework. To predict when to stop active learning procedure, it is logical to consider the changes of accuracy performance of the classifier as a signal to stop the learning iteration. In other words, during the range predicted by the proposed solution, if the change of accuracy performance of the learner (classifier) is very small, we could assume that the current classifier has reached maximum effectiveness.

8 Conclusion and Future Work

In this paper, we consider the class imbalance problem in WSD tasks, and analyze the effect of resampling techniques including over-sampling and under-sampling in active learning. Experimental results show that over-sampling is a relatively good choice in active learning for WSD in highly imbalanced data. Under-sampling causes negative effect on active learning. A new over-sampling method named BootOS based on bootstrap technique is proposed to alleviate the within-class imbalance problem of over-sampling, and works better than ordinary over-sampling in active learning for WSD. It is noteworthy that none of these techniques require to modify the architecture or learning algorithm; therefore, they are very easy to use and extend to other applications. To predict when to stop active learning, we adopt two strategies including max-confidence and min-error as stopping conditions. According to our experimental results, we suggest a prediction solution by considering max-confidence as the upper bound and min-error as the lower bound of stopping conditions for active learning.

In the future work, we will study how to exactly identify these borderline samples thus they are not firstly selected in active learning procedure. The borderline samples have the higher entropy values meaning least confident for the current classifier. The borderline instances can be detected using the concept of *Tomek links* (Tomek 1976). It is also worth studying cost-sensitive learning for active learning with imbalanced data, and using such techniques for WSD.

References

A. L. Berger, S. A. Della, and V. J Della. 1996. *A maximum entropy approach to natural language processing*. Computational Linguistics 22(1):39–71.

E Brill. 1992. *A simple rule-based part of speech tagger*. In the Proceedings of the Third Conference on Applied Natural Language Processing.

Y. S. Chan and H. T. Ng. 2006. *Estimating class priors in domain adaptation*. In Proc. of ACL06.

N. Chawla, K. Bowyer, L. Hall, W. Kegelmeyer. 2002. *SMOTE: synthetic minority over-sampling technique*. Journal of Artificial Intelligence Research, 2002(16): 321-357

J. Chen, A. Schein, L. Ungar, M. Palmer. 2006. *An empirical study of the behavior of active learning for word sense disambiguation*. In Proc. of HLT-NAACL06

I. Dagan, O. Glickman, A. Gliozzo, E. Marmorshtein, and C. Strapparava. 2006. *Direct Word Sense Matching for Lexical Substitution*. In Proc. of ACL'06

H. T. Dang and M. Palmer. 2005. *The Role of Semantic Roles in Disambiguating Verb Senses*. In Proc. of ACL'05.

A. Estabrooks, T. Jo and N. Japkowicz. 2004. *A multiple resampling method for learning from imbalanced data set*. Computational Intelligence, 20(1):18-36

Y. Hamamoto, S. Uchimura and S. Tomita. 1997. *A bootstrap technique for nearest neighbor classifier design*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(1):73-79

V. Hoste, A. Kool, and W. Daelemans. 2001. *Classifier optimization and combination in the English all words task*. In Proc. of the SENSEVAL-2 workshop

E. Hovy, M. Marcus, M. Palmer, L. Ramshaw and R. Weischedel. 2006. *Ontonotes: The 90% Solution*. In Proc. of HLT-NAACL06.

N. Ide and J. Veronis. 1998. *Introduction to the special issue on word sense disambiguation: the state of the art*. Computational Linguistics, 24(1):1-37

N. Japkowicz and S. Stephen. 2002. *The class imbalance problem: a systematic study*. Intelligent Data Analysis, 6(5):429-450

T. Jo and N. Japkowicz. 2004. *Class imbalances versus small disjuncts*. SIGKSS Explorations, 6(1):40-49

U. S. Kohomban and W. S. Lee. 2005. *Learning Semantic Classes for Word Sense Disambiguation*. In Proc. of ACL'05

M. Kubat and S. Matwin. 1997. *Addressing the curse of imbalanced training sets: one-sided selection*. In Proc. of ICML97

Y.K. Lee and. H.T. Ng. 2002. *An empirical evaluation of knowledge sources and learning algorithm for word sense disambiguation*. In Proc. of EMNLP-2002

D. D. Lewis and W. A. Gale. 1994. *A sequential algorithm for training text classifiers*. In Proc. of SIGIR-94

D.D. Lewis and J. Catlett. 1994. *Heterogeneous uncertainty sampling for supervised learning*. In Proc. of ICML94

M. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. *Building a large annotated corpus of English: the Penn Treebank*. Computational Linguistics, 19(2):313-330

A. McCallum and K. Nigram. 1998. *Employing EM in pool-based active learning for text classification*. In Proc. 15th ICML

D. McCarthy, R. Koeling, J. Weeds, and J. Carroll. 2004. *Finding predominant senses in untagged text*. In Proc. of ACL04

I. Muslea, S. Minton, and C. A. Knoblock. 2000. *Selective sampling with redundant views*. In Proc. of National Conference on Artificial Intelligence

I. Tomek. 1976. *Two modifications of CNN*. IEEE Transactions on Systems, Man and Cybernetics, 6(6):769-772

G. M. Weiss. 2004. *Mining with rarity – problems and solutions: a unifying framework*. SIGKDD Explorations, 6(1):7-19

N. Xue, J. Chen and M. Palmer. 2006. *Aligning Features with Sense Distinction Dimensions*. In Proc. of ACL'06

Z. Zhou, X. Liu. 2006. *Training cost-sensitive neural networks with methods addressing the class imbalance problem.* IEEE Transactions on Knowledge and Data Engineering, 18(1):63-77

L. Zhang, J. Zhu, and T. Yao. 2004. *An evaluation of statistical spam filtering techniques*. ACM Transactions on Asian Language Information Processing, 3(4):243–269.

C. Zhang and T. Chen. 2002. *An active learning framework for content-based information retrieval*. IEEE Transactions on Multimedia, 4(2):260-268