ACTIVE LEARNING FOR SPOKEN LANGUAGE UNDERSTANDING

Gokhan Tur *Robert E. Schapire** *Dilek Hakkani-Tür*

AT&T Labs – Research
180 Park Avenue, Florham Park, NJ 07932 USA
{gtur,schapire,dtur}@research.att.com

ABSTRACT

In this paper, we describe active learning methods for reducing the labeling effort in a statistical call classification system. Active learning aims to minimize the number of labeled utterances by automatically selecting for labeling the utterances that are likely to be most informative. The first method, inspired by certainty-based active learning, selects the examples that the classifier is least confident about. The second method, inspired by committee-based active learning, selects the examples that multiple classifiers do not agree on. We have evaluated these active learning methods using a call classification system used for AT&T customer care. Our results indicate that it is possible to reduce human labeling effort at least by a factor of two.

1. INTRODUCTION

Voice-based natural dialog systems enable customers to express what they want in spoken natural language. Such systems automatically extract the meaning from speech input and act upon what people actually say, in contrast to what one would like them to say, shifting the burden from users to the machine [1]. In a natural spoken dialog system, identifying the customer's intent can be seen as a call classification problem.

When statistical classifiers are employed for this purpose, they are trained using large amounts of task data which is transcribed and labeled by humans, a very expensive and laborious process. By "labeling," we mean assigning a predefined call type to each utterance. Building better call classification systems in a shorter time frame motivates us to employ *active learning* techniques. We aim to reduce the number of training examples to be labeled by inspecting the unlabeled examples, and intelligently selecting the most *informative* ones with respect to a given cost function for a human to label [2]. The goal of the active learning algorithm is to select the examples which will have the largest improvement on the performance, hence reduce the amount of human labeling effort.

*Currently at the Department of Computer Science, Princeton University, Princeton, NJ, 08544.

Selectively sampling the utterances assumes that, there is a pool of candidate utterances to label. In a deployed natural dialog system, this is indeed the case, where a constant stream of raw data is collected from the field. Then the aim of active learning is to come up with a sorting algorithm for these utterances, hopefully indicating their informativeness. The intuition is that there is a reverse correlation with the confidence of the classifier and the informativeness of that utterance. That is, the higher the classifier's confidence, the less informative an utterance. We can expect that the classifier would be trained better if we do label the utterances which are different enough for the classifier.

In the following section, we present briefly the active learning literature, as well as review some of the related work in language processing. In Section 3, we describe our algorithms, and in Section 4 we present our experiments and results.

2. RELATED WORK

The search for effective training data sampling algorithms, in order to have better systems with less annotated data by giving the system some control over the inputs on which it trains, has been studied under the title of active learning.

Previous work in active learning has concentrated on two approaches: certainty-based methods and committee-based methods. In the *certainty-based methods*, an initial system is trained using a small set of annotated examples [3]. Then the system examines and labels the unannotated examples and determines the certainties of its predictions on them. The k examples with the lowest certainties are then presented to the labelers for annotation.

In the *committee-based methods*, a distinct set of classifiers is also created using the small set of annotated examples [2, 4]. The unannotated instances whose annotations differ most when presented to different classifiers are presented to the labelers for annotation. In both paradigms, a new system is trained using the new set of annotated examples, and this process is repeated until the system performance converges to a limit.

Active learning has previously been applied to support-

vector machines [5, 6]. In the language processing framework, certainty-based methods have been used for natural language parsing and information extraction [7, 8] and word segmentation [9]. In our previous work, we have presented an active learning approach for automatic speech recognition [10].

3. APPROACH

In this study we have tried two active learning methods. In both methods, we assumed that the candidate utterances are first recognized by the same automatic speech recognizer (ASR), so we deal with only text input of the same quality.

First, inspired by the certainty-based active learning methods, we select for labeling the examples that we predict the classifier is most unsure about, and leave out the ones that it has classified with high confidence.

We first train a classifier using a small set of labeled data S_t. This approach is independent of the classifier used. Using this classifier, we classify the utterances that are candidates for labeling S_u. We then use the classifier score to predict which candidates are classified with high/low confidence. We transcribe the utterances that are most likely to have classification errors. Our algorithm is as follows:

1. Begin with a small amount of training data S_t, and a larger amount of unlabeled data in the pool S_u
2. While (labelers/utterances are available) do
   - 2.1 Train a classifier using the current training data S_t
   - 2.2 Classify the utterances in the pool S_u using this classifier and compute the call type confidence scores for all utterances
   - 2.3 Sort the candidate utterances with respect to the score of the maximum scoring call type
   - 2.4 Select the lowest scored k utterances from S_u and label them. Call the new labeled set S_i
   - 2.5 $S_t = S_t \bigcup S_i$; $S_u = S_u - S_i$

The parameter k is mainly determined by the capacity of the manual labeling effort. The other parameter is the sorting criterion. One can come up with a different sorting criterion for sorting the utterances, such as the difference of top two call type scores, or by using other or all the call type scores. It is also possible to make a "cheating" experiment using the score of the true call type for sorting.

Note that the distribution of the call types in the selectively sampled training data have skewed from their priors. That is the distribution of call types has become different in training and test data. One solution is to adjust the priors by up-sampling the data. In our experiments we have simply ignored this problem.

As a second method, inspired by committee-based active learning methods, we select the examples that multiple classifiers disagree on the most and leave out the ones on which there is agreement (even if their score is low). Here is the algorithm:

1. Begin with a small amount of training data S_t and a larger amount of unlabeled data in the pool S_u
2. While (labelers/utterances are available) do
   - 2.1 Train multiple classifiers independently using the current training data S_t
   - 2.2 Classify the utterances in the pool S_u using these classifiers and compute the call type confidence scores for all utterances
   - 2.3 Sort the candidate utterances with respect to the score of the maximum scoring call type according to one of the classifiers if the classifiers disagree
   - 2.4 Select the lowest scored k utterances from S_u, and label them. Call the new labeled set S_i
   - 2.5 $S_t = S_t \bigcup S_i$; $S_u = S_u - S_i$

Note that getting a low score is not enough to get selected; it is also necessary that classifiers disagree. It is also possible to modify this criterion, such as sorting using the (weighted) multiplication of the scores of the top choices. A "cheating" experiment is also possible in this case by using the true labels as the output of a perfect classifier which does not make any mistakes. This approach is independent of the classifiers used, but it makes sense to use different kinds of classifiers which have comparable performance using different feature sets.

4. EXPERIMENTS AND RESULTS

We have evaluated these active learning methods using the utterances from the database of the *How May I Help You?*$^{\mathrm{SM}}$ system for AT&T customer care [11, 12]. In this natural dialog system, users are asking questions about their phone bills, calling plans, etc., and the system aims to classify them into 49 call types in total, such as *Billing Credit*, or *Calling Plans*.

We performed two sets of experiments. In the first set, we used 21,953 utterances from the responses to just the greeting and specification prompts. We used 17,553 of them for training, and 4,400 of them for testing. We used the Llama [13] support vector machine (SVM) classifier [14] in this test as the basic classifier.

First, we checked the informativeness of using the classifier score with this data. We trained an SVM classifier using all the data. We used word n-grams as features for classification. For each classifier score bin, we computed the accuracy of its decision. As seen in Figure 3, we got an almost diagonal curve, as expected.

We began with a test of our first method using only an SVM classifier. In order to see the actual improvement, we

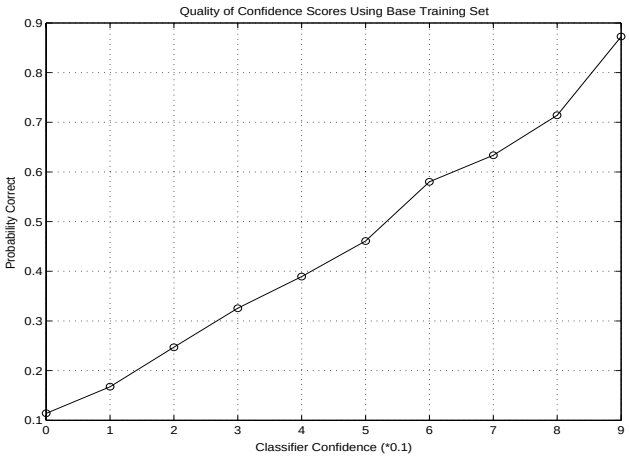

Fig. 1. Accuracy with respect to the classifier score.

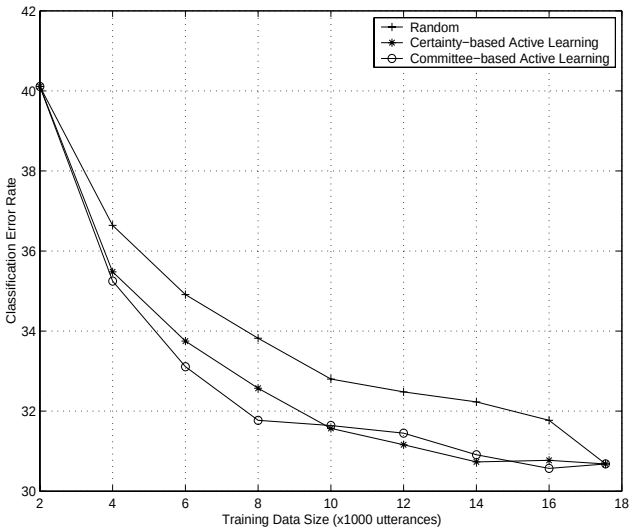

Fig. 2. The error rates using random and active learning methods for labeling.

performed controlled experiments comparing our methods with random sampling. We incrementally trained the classifier every 2000 utterances ($k = 2000$) and generated learning curves for classification error rate, one for random sampling and one for selective sampling, which are presented in Figure 4. We define the classification error rate as the ratio of the utterances where the maximum scoring call type is not one of the true call types. It is evident that selective sampling significantly reduces the need for labeled data. For instance, achieving an error rate of 32% requires roughly 15,000 random examples but only about 9,000 selectively sampled examples, a 40% savings in labeling effort. The accuracy of the classifier improves much faster than using random sampling. Note that the final 2,553 utterances decrease the error rate by around 2% if we use random sampling. In the case of selective sampling, they have basically no effect on classification performance, another indicator that the certainty-based method works well in determining the informativeness of an utterance.

We used the same data to test the effect of our second approach. In addition to the SVM classifier, we trained another classifier using BoosTexter [15], an implementation of the AdaBoost algorithm, again using word n-grams as features. We again generated learning curves, one for random sampling and the other for selective sampling as seen in Figure 4. We see that this method outperformed the previous method, and we managed to achieve the same performance obtained using 16,000 random utterances with only 8,000 selectively sampled utterances, a factor of two reduction in labeling effort.

Note that, because these are controlled experiments, the performance using all available training data is the same both for random sampling and selective sampling. Therefore, eventually the selective sampling curve must reflect the "uninformative" data at the end of the list. For this reason, in the actual implementation, one may expect the performance improvement using selective sampling to be larger.

In these experiments, we have left aside the responses for confirmation prompts, such as "yes" or "no", and some machine-initiated sub-dialog prompts, such as "*What is your phone number?*" In order to see the performance with all of the data we conducted a similar experiment following the certainty-based approach using BoosTexter alone. In this experiment, we used the a larger dataset of 51,755 utterances, including the easy-to-classify examples omitted in the other experiments. We used 40,000 randomly selected examples for training (i.e., as potential candidates for labeling), and the remainder for testing. The procedure was the same as for SVM's except that the confidence of the classifier in its prediction on a given example was defined to be the difference between the scores of the top two classes. Thus, on each iteration of the active-learning loop, the examples for which this difference was smallest were added to our training pool of labeled utterances.

Figure 5 shows the results. In this experiment, we added 500 examples ($k = 500$) on each iteration. Also, the entire experiment was repeated ten times and the results averaged. As before, we see substantial improvements using active learning. For instance, achieving a test error of 25% requires 40,000 examples if randomly chosen, but only 13,000 actively selected examples, a savings of 68%.

The active learning curve also exhibits an interesting phenomenon previously noted by Schohn and Cohn [5] for SVM's: better performance can actually be achieved using *fewer* training examples. Apparently, examples added toward the end are not only uninformative, but actually disinformative.

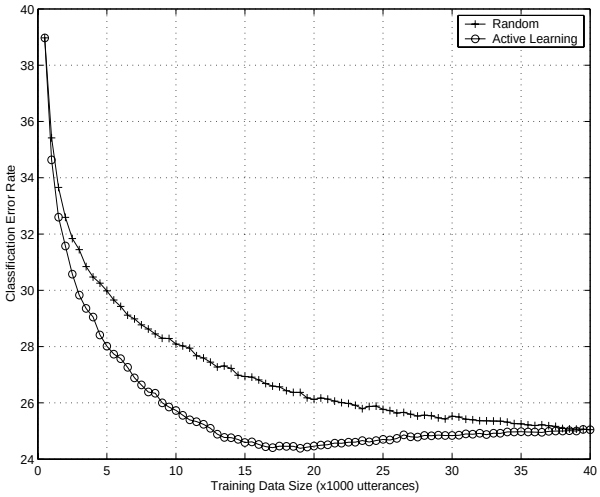

Fig. 3. The error rates achieved by BoosTexter using active and random selection of examples for labeling.

5. CONCLUSIONS

We have presented active learning algorithms for reducing the number of labeled training examples by selectively sampling a subset of the unlabeled data. We have shown that, for the task of call classification, using selective sampling it is possible to speed up the learning rate of the classifier with respect to the amount of labeled utterances. Our results indicate that we have managed to achieve the same call classification accuracy using less than half labeled data. We have tried two approaches, one inspired by certainty-based, the other by committee-based active learning methods. We have seen that using the latter approach helps more, although we have used two large margin classifiers, SVM's and AdaBoost, with the same features sets, namely, word n-grams. One may expect better results with different kinds of classifiers using different kinds of information to extract features.

6. ACKNOWLEDGMENTS

We would like to thank Patrick Haffner for providing the LLAMA-SVM classifier software and for many helpful discussions. Thanks also to Mazin Rahim for numerous helpful suggestions.

7. REFERENCES

[1] A. L. Gorin, G. Riccardi, and J. H. Wright, "Automated natural spoken dialog," *IEEE Computer Magazine*, vol. 35, no. 4, pp. 51–56, April 2002.

[2] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine Learning*, vol. 15, pp. 201–221, 1994.

[3] D. D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *Proceedings of the ICML*, 1994.

[4] I. Dagan and S. P. Engelson, "Committee-based sampling for training probabilistic classifiers," in *Proceedings of the ICML*, 1995.

[5] G. Schohn and D. Cohn, "Less is more: Active learning with support vector machines," in *Proceedings of the ICML*, 2000.

[6] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2001.

[7] C. Thompson, M. E. Califf, and R. J. Mooney, "Active learning for natural language parsing and information extraction," in *Proceedings the ICML*, 1999.

[8] M. Tang, X. Luo, and S. Roukos, "Active learning for statistical natural language parsing," in *Proceedings the ACL*, 2002.

[9] M. Sassano, "An empirical study of active learning with support vector machines for japanese word segmentation," in *Proceedings the ACL*, 2002.

[10] D. Hakkani-Tür, G. Riccardi, and A. Gorin, "Active learning for automatic speech recognition," in *Proceedings of the ICASSP*, 2002.

[11] J. Wright, A. Gorin, and G. Riccardi, "Automatic acquisition of salient grammar fragments for call-type classification," in *Proceedings of the Eurospeech*, Rhodes, Greece, September 1997.

[12] G. Tur, J. Wright, A. Gorin, G. Riccardi, and D. Hakkani-Tür, "Improving spoken language understanding using word confusion networks," in *Proceedings of the ICSLP*, 2002.

[13] P. Haffner, "Llama – General software library for large margin classifiers," http://www.research.att.com/∼haffner/llama.

[14] V. N. Vapnik, *Statistical Learning Theory*, John Wiley and Sons, New York, 1998.

[15] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine Learning*, vol. 39, no. 2/3, pp. 135–168, 2000.