ACTIVE LABELING FOR SPOKEN LANGUAGE UNDERSTANDING

Gokhan Tur *Mazin Rahim* *Dilek Hakkani-Tür*

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

ABSTRACT

State-of-the-art spoken language understanding (SLU) systems are trained using human-labeled utterances, preparation of which is labor intensive and time consuming. Labeling is an error-prone process due to various reasons, such as labeler errors or imperfect description of classes. Thus, usually a second (or maybe more) pass(es) of labeling is required in order to check and fix the labeling errors and inconsistencies of the first (or earlier) pass(es). In this paper, we check the effect of labeling errors for statistical call classification and evaluate methods of finding and correcting these errors by checking minimum amount of data. We describe two alternative methods to speed up the labeling effort, one is based on the confidences obtained from a prior model and the other completely unsupervised. We call the labeling process employing one of these methods as *active labeling*. Active labeling aims to minimize the number of utterances to be checked again by automatically selecting the ones that are likely to be erroneous or inconsistent with the previously labeled examples. Although very same methods can be used as a postprocessing step to correct labeling errors, we only consider them as part of the labeling process. We have evaluated these active labeling methods using a call classification system used for AT&T natural dialog customer care system. Our results indicate that it is possible to find about 90% of the labeling errors or inconsistencies by checking just half the data.

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 one or more predefined label(s) (*call type(s)*) to each utterance. It is clear that the bottleneck in building a decent statistical system is the time spent for high quality labeling. In order to achieve an acceptable quality, each one of the labels is usually verified by an independent party, since it is a very error-prone process. An utterance is mislabeled mostly because of two reasons: The first one is simply the labeler error, and the second one is the imperfect description of classes. Also note that, for the multi-label tasks, where an utterance may get more than one label, it is necessary to label them all. If any of the labels is missing, it is considered as a labeling error. Thus, usually a second (or maybe more) pass(es) of labeling is required in order to check and fix the labeling errors and inconsistencies of the first (or earlier) pass(es). The motto "There is no data like more data" holds better if the data is less "noisy", i.e. contains less than tolerable number of mislabeled utterances. Most state-of-the-art classifiers tolerate a few percentage points of noisy data, but more errors than that ruins the classification performance, even though how robust the classifiers are.

Building better call classification systems in a shorter time frame motivates us to develop novel techniques. One solution would be focusing on active learning to train decent models using less data [2, 3]. Another solution, which we present in this paper, is reducing the labeling time to get decent training data. By employing active labeling during the labeling process, we aim at decreasing the number of training examples to be checked in the second (or latter) pass(es) by automatically selecting the ones that are likely to be erroneous or inconsistent with previously labeled examples, hence reduce the amount of human labeling effort.

In the following section, we 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

Most related work deals with the problem of removing labeling errors as a postprocessing step. Those studies assume

that the labeling process is over and the aim is to detect a relatively small number of errors in the data. The difference of our work is that, we aim at proposing a subset of utterances which need to be checked again during the labeling process, although the same methods may be used for postprocessing the already labeled data.

Abney *et al.* have presented a method for automatic detection of labeling errors [4]. This study, specific to the AdaBoost classification algorithm [5], uses some specific characteristics of the Adaboost algorithm for this purpose. This algorithm assigns importance weights to utterances during training depending on whether they are classified correctly or not. Since mislabeled examples tend to be hard examples to classify correctly, they tend to have large weights.

Eskin has proposed a method using anomaly detection, which is used to determine which elements of a large data set do not conform to the whole [6]. This method, originated from computer security, is applied to part-of-speech tagging problem using the Penn Treebank corpus.

Murata *et al.* have presented a method for detecting and correcting annotation errors in a modality corpus used for machine translation [7]. Their method, similar to the certainty-based approach of ours depends on the confidences obtained from the statistical system trained using the corpora that they want to correct.

van Halteren has proposed a method in order to detect the inconsistencies in a manually tagged corpus [8].

Other studies are either related to making the classifier robust to noisy data [9, among others] or they briefly mention that they have manually checked the data and corrected labeling errors [10, among others].

3. APPROACH

In this study we propose two alternative active labeling methods. Both methods assume that, we have a set of labeled (but not checked) utterances, probably containing some amount of errors and inconsistencies. The first one also assumes a readily available prior model trained with human-labeled and checked data. The second is completely unsupervised and does not need any prior model. Both methods are independent of the classifier used as long as some confidence scores are output.

3.1. Certainty-Based Active Labeling

In this method, inspired by the certainty-based active learning methods [11, 2, 3], for checking, we select the examples that the classifier is confident about but disagree with the first labeler's decision, and leave out the ones that the classifier agrees with the labeler's decision with high confidence. This method requires that the classifier returns a confidence, $Q(i|U)$, between 0 and 1 for each of the labels, $i \in L$, where L is the set of all calltypes, for a given utterance, U. Indeed this is the case for most statistical classifiers.

This method assumes that there exists a previously trained classifier (probably using the previous portions of training data) and a set of utterances which have been labeled (but not checked). Using this prior classifier, we classify those candidate utterances. We then use the classifier confidence score to predict which candidate utterances are classified with high/low confidence. Once we have the classifier confidence for all the call types and the first labeler's decision, we can come up with various criteria for sorting the candidate utterances for checking. For example, it is possible to select the ones where classifier's top choice is not among the call types labeler has selected. This criterion works fine for most cases, but misses one type of errors for multi-label tasks: It is sometimes possible for the second pass labeler to add an additional call type to that utterance. Even though classifier's top choice matches one of the labels of the first pass with high enough confidence, this does not mean that this utterance has been labeled correctly. Alternatively, one can select the ones where first pass labeler's some or all choices get some confidences more than some threshold. Similar to the previous criterion, this one is also not enough. There are cases where there is another calltype which gets even more confidence and should be added to the true calltypes.

As seen, it is necessary to consider all the confidences of all the calltypes. Considering these issues, we have come up with a more general selection and sorting criterion: the Kullback-Leibler (KL) divergence (or binary relative entropy) between the first pass labels, P, and the classifier outputs, Q. More formally, we compute:

$$KL(P \parallel Q) = \sum_{i \in L} p_i \times log(\frac{p_i}{q_i}) + 1(\quad - p_i) \times log(\frac{1 - p_i}{1 - q_i})$$

where L is the set of all calltypes. q_i is the probability of the ith calltype obtained from the classifier. Since the first pass labels is not a probability distribution, we handled it as follows: we set p_i to 1 if that call type is labeled in the first pass and 0 otherwise.

This method can be summarized as follows:

- Using the prior model, classify all the utterances. That means getting confidences for all the labels for all the utterances.
- Check only the utterances where the Kullback-Leibler divergence is more than some threshold.

3.2. Unsupervised Active Labeling

As the second method, we assume the case where there is no readily available classifier model. That is, only the set of utterances with a some amount of errors and inconsistencies is available. In such a case, we propose a different active labeling method: Use the set of candidate utterances

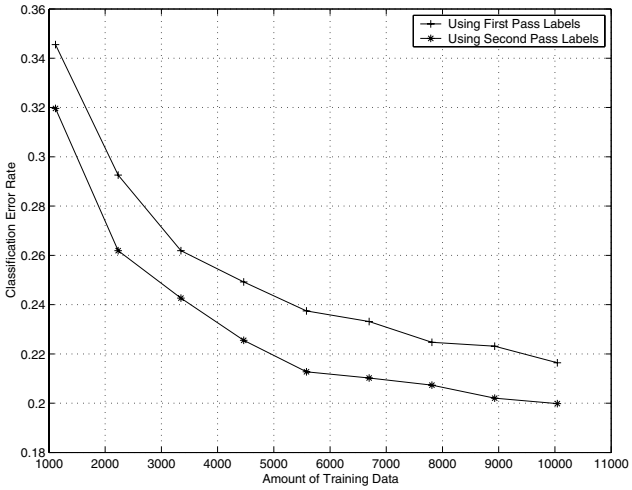

Fig. 1. The effect of labeling errors to classification performance. Classifier achieves the same performance using half the amount of corrected data compared to noisy data.

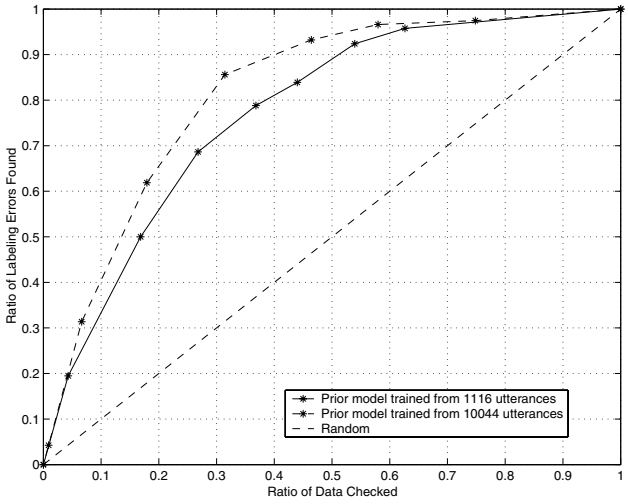

Fig. 2. The ratio of labeling errors found with respect to the ratio of utterances checked using certainty-based active labeling. The baseline performance, i.e. with no active labeling, is the diagonal, where both ratios should be equal.

as if it is the training data, and train the classifier. Then classify the same noisy data. The motivation is that, the utterances in the training data which are not classified correctly are more probably the labeling errors. This method is similar to the work described in [4]. The difference is that this is not specific to a particular classification algorithm. Since this method does not require any human-labeled high quality data, we call this as *unsupervised* active labeling.

Similar to the certainty-based method, it is also to possible to put a threshold if the classification algorithm is iterative, such as boosting [5]. In such cases, the classifier may continue training with noisy data until the error rate for the training data is less than some threshold, and the utterances which are not classified as in their first pass labels are sent for a second pass of labeling. Alternatively, one may check the confidences of the labels and check the ones which are classified with a low confidence, similar to the certainty-based method.

4. EXPERIMENTS AND RESULTS

We have evaluated these active labeling methods using the utterances from a natural dialog customer care system. This system aims to classify the user utterances into 32 call types in total. In our experiments we used a set of 11,160 utterances, and split 90% of them for training and 10% of them for testing. In total, 11% of the utterances have more than one label, and there are 1.15 labels per utterance on the average. In all the experiments, we used Boostexter as the classifier [5] and n-grams of the utterances as features.

Before implementing any of the active labeling methods, we have checked the effect of the labeling errors to the classification performance. For this purpose, we trained the classifier using the first pass labels and second pass corrected labels, and checked the difference. 13% of the utterances are corrected in our test data, and 9% of them are changed completely, that is there is no common label left between the first and second passes. This is a large human error rate and motivates that a second pass of checking is crucial to train decent classifier models. Note that probably there are some more labeling errors even after the second pass. Figure 1 shows the classification performances using checked and unchecked training data. As seen, using unchecked labels, the classification error rate increases by 2%-3% points absolute, that is about 10% relative reduction in the performance. In other words, the classifier needs twice as much unchecked data in order to obtain the same performance with checked data. These results justify the motivations for active labeling.

Figure 2 shows the results of the experiments using the certainty-based method. It draws the ratio of labeling errors found with respect to the ratio of utterances checked. The diagonal dashed line is the baseline where both ratios are equal. This is the performance you may expect without active labeling. We have drawn these curves by putting a threshold on the KL divergence. The solid one is obtained using a prior classification model trained using 1,116 utterances and dashed curve using all 10,044 utterances. We have not drawn the curves for prior model data sizes between these, since they lie in-between, as expected. For both curves, this method outperforms the baseline, even using just 1,116 utterances and finds about 90% of the errors using just half the data, or finds 75% of the errors checking one third of the utterances. Furthermore, the active labeling

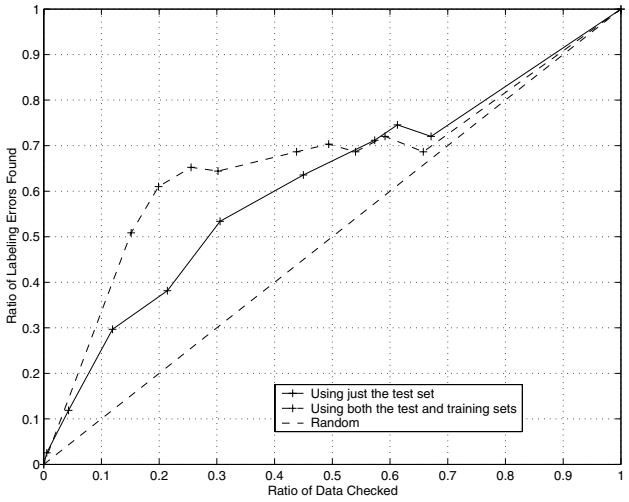

Fig. 3. The ratio of labeling errors found with respect to ratio of utterances checked using unsupervised active labeling. Active labeling still works even using only noisy data.

performance increases as the prior model gets better with more data. The ratio of labeling errors found increases from 72% to 83% by using a better prior model when 30% of the utterances are checked.

Figure 3 shows the results of the experiments using the unsupervised method. We have drawn two curves by different numbers of Boosting iterations. The solid one is obtained using just the test data. The dashed curve is obtained by using all 11,160 utterances, but then evaluating only on the test set. This method outperforms the baseline, but underperforms the certainty-based method. It finds about 70% of the errors using just half the data, or finds about 2/3 of the errors checking 1/3 of the utterances. In order to see the effect of number of the candidate utterances used in this method to the performance, we have varied the candidate utterance set size, but only checked the performance on the test set to get comparable results. At 30% of the data checked, the ratio of labeling errors found increases about 10% absolute using more number of utterances.

5. CONCLUSIONS

We have presented active labeling algorithms for reducing the number of utterances to be checked by automatically selecting the ones that are likely to be erroneous or inconsistent with the previously labeled examples. We have shown that, for the task of call classification, using active labeling it is possible to speed up the second pass of labeling significantly. Our results indicate that we have managed to find about 90% of the labeling errors using just half the data. These results are especially important when there is little time for noise-free labeling. It is also clear that these methods can also be used to clean up and even correct already labeled data as a postprocessing step. The first method is especially very suitable for this purpose. Furthermore, these methods are pretty general, and can be used for any classification task; it is not limited to only call classification tasks.

6. 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. Hakkani-Tür, G. Riccardi, and A. Gorin, "Active learning for automatic speech recognition," in *Proceedings of the ICASSP*, Orlando, FL, May 2002.

[3] G. Tur, R. E. Schapire, and D. Hakkani-Tür, "Active learning for spoken language understanding," in *Proceedings of the ICASSP*, Hong Kong, China, May 2003.

[4] S. Abney, R. Schapire, and Y. Singer, "Boosting applied to tagging and PP attachment," in *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

[5] 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.

[6] E. Eskin, "Detecting errors within a corpus using anomaly detection," in *Proceedings of the NAACL*, Seattle, WA, April 2000.

[7] M. Murata, M. Utiyama, K. Uchimoto, Q. Ma, and H. Isahara, "Correction of errors in a modality corpus used for machine translation using machine-learning," in *Proceedings of the TMI*, Japan, March 2002.

[8] H. van Halteren, "The detection of inconsistency in manually tagged text," in *Proceedings of the Workshop on Linguistically Interpreted Corpora*, Luxembourg, August 2000.

[9] M. Kearns, "Efficient noise-tolerant learning from statistical queries," in *Proceedings of the ACM Symposium on Theory of Computing*, 1993.

[10] I. Hendrickx, A. van der Bosch, V. Hoste, and W. Daelemans, "Dutch word sense disambiguation," in *Proceedings of the Workshop on Word Sense Disambiguation*, Philadelphia, PA, July 2002.

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