MODEL ADAPTATION FOR SPOKEN LANGUAGE UNDERSTANDING

Gokhan Tur

AT&T Labs – Research
Florham Park, NJ 07932 USA
gtur@research.att.com

ABSTRACT

In this paper, we present a novel adaptation method for intent classification using Boosting in a spoken language understanding system. The goal is adapting an existing model to a new target application, which is similar but may have different intents or intent distributions. Adaptation can also be employed for a single application where the intent distribution varies by time. We assume the target application has a small amount of labeled data. We also propose employing active learning to selectively sample the data to label for adaptation. Our results indicate that we can achieve the same intent classification accuracy using less than half of the labeled data when there is not much training data available. Furthermore, combined with active learning we see 18.6% relative reduction in classification error rate.

1. INTRODUCTION

Spoken dialog systems aim to identify intents of humans, expressed in natural language, and take actions accordingly, to satisfy their request. The intent of the speaker is identified using a natural language understanding component. This step can be seen as a classification problem [1, 2]. In this study, we have used a Boosting-style classification algorithm [3]. As an example, consider the utterance *I would like to know my account balance*, in a customer care application from a financial domain. Assuming that the utterance is recognized correctly, the corresponding intent would be *Request(Balance)* and the action would be telling the balance to the user after prompting for the account number or routing this call to the billing department.

Data-driven classifiers are trained using large amounts of task data which is usually transcribed and then labeled by humans, an expensive and laborious process. By "labeling", we mean assigning one or more of the predefined intents to each utterance.

In this paper, we present a supervised adaptation method for natural language intent classification. When many spoken dialog systems using similar intent classification models are needed to be built in a shorter time frame, the new target application can be bootstrapped only by labeling a small amount of data using adaptation techniques. The target application may include intents which are already mostly covered by an existing application, but maybe with different prior distributions. For instance, consider a new application from the same domain for customer care applications. Adaptation can be employed for statistical models where the target application does not match the training data. This may include continuous adaptation of an existing model to time varying statistics or exploiting out of domain data for training the target model.

Although statistical model adaptation has been a well studied area in speech recognition for acoustic and language modeling [4, 5, 6], there is comparably less work done on natural language processing. One recent study is on the adaptation of natural language understanding using a common adaptation method of *maximum a posteriori* (MAP) adaptation [7], which adapts the hidden vector state model built for ATIS application to DARPA Communicator. Another study is about supervised and unsupervised adaptation of probabilistic context free grammars to a new domain using again MAP adaptation [8].

In the following section, we briefly explain boosting algorithms. Then, in Section 3, we propose a new adaptation method. We conclude after presenting the experiments and results.

2. BOOSTING

Boosting is an iterative procedure; on each iteration, t, a weak classifier, h_t is trained on a weighted training set, and at the end, the weak classifiers are combined into a single, combined classifier. The algorithm generalized for multi-class and multi-label classification is given in Figure 1. Let $\mathcal{X}$ denote the domain of possible training examples and let $\mathcal{Y}$ be a finite set of classes of size $|\mathcal{Y}| = k$. For $Y \subseteq \mathcal{Y}$, let $Y[l]$ for $l \in \mathcal{Y}$ be

$$Y[l] = \begin{cases} +1 & \text{if } l \in Y \\ -1 & \text{otherwise.} \end{cases}$$

The algorithm begins by initializing a uniform distribution $D_1(i, l)$ over training examples i and labels l. After each round this distribution is updated so that the example-class combinations which are easier to classify get lower weights

- Given training data from the instance space $S = \{(x_1, Y_1), ..., (x_m, Y_m)\}$ where $x_i \in \mathcal{X}$ and $Y_i \subseteq \mathcal{Y}$.
- Initialize the distribution $D_1(i, l) = 1/mk$.
- For each iteration $t = 1, ..., T$ do
  - Train a base learner h_t using distribution D_t.
  - Update
$$D_{t+1}(i, l) = \frac{D_t(i, l)e^{-\alpha_t Y_i[l]h_t(x_i,l)}}{Z_t}$$
where Z_t is a normalization factor and α_t is the weight of the base learner.
- Output the final classifier defined as:
$$f(x, l) = \sum_{t=1}^{T} \alpha_t h_t(x, l).$$

Fig. 1. The algorithm *Adaboost.MH*.

and vice versa. The intended effect is to force the weak learning algorithm to concentrate on the examples and labels that will be the most beneficial to the overall goal of finding a highly accurate classification rule.

This algorithm can be seen as a procedure for finding a linear combination of base classifiers which attempts to minimize an exponential loss function [9], which in this case is:
$$\sum_i \sum_l e^{-Y_i[l]f(x_i,l)}.$$
An alternative would be to minimize a logistic loss function as suggested by [10], namely
$$\sum_i \sum_l \ln(1 + e^{-Y_i[l]f(x_i,l)}).$$
In that case, the confidence of a class, l, for an example, x_i is computed as:
$$P(Y_i[l] = +1|x_i) = \frac{1}{1 + e^{-f(x_i,l)}}$$
A more detailed explanation and analysis of this algorithm can be found in [3]. In our experiments, we used the BoosTexter tool, which is an implementation of the Boosting algorithm [1]. For text categorization, BoosTexter uses word n-grams as features, and each weak classifier (or "decision stump") checks the absence or presence of a feature.

3. APPROACH

In this work, the aim is to exploit the existing labeled data and models for boosting the performance of the new similar applications using a supervised adaptation method. The basic assumption is that there is an intent model trained with data similar to the target application. Then the idea is adapting this classification model using the small amount of already labeled data from the target application, thus reducing the amount of human-labeling effort necessary to come up with decent statistical intent classification systems. The very same adaptation technique can be employed for improving the existing model for non-stationary new data.

There are at least two other ways of exploiting the existing labeled data from a similar application. We will evaluate and compare these methods to adaptation in the next section.

- **Simple Data Concatenation** (*simple*)**:** where the new classification model is trained using the data from the previous application concatenated to the data labeled for the target application.
- **Tagged Data Concatenation** (*tagged*)**:** where the new classification model is trained using both data sets, but each set is tagged with the source application. That is, in addition to the utterances, we use the source of that utterance as an additional feature during classification.

3.1. Classification Model Adaptation

For adaptation, we begin with an existing classification model. Then using the labeled data from the target application we build a new model based on this existing one. This method is similar to incorporating prior knowledge or exploiting unlabeled utterances for Boosting [11, 12]. In those works, a model which fits both the training data and the task knowledge or machine labeled data is trained. In our case, the aim is to train a model that fits both a small amount of application specific labeled data and the existing model from a similar application. More formally the Boosting algorithm tries to fit both the newly labeled data and the prior model using the following loss function:
$$\sum_i \sum_l (\ln(1+e^{-Y_i[l]f(x_i,l)}) + \eta KL(P(Y_i[l] = 1|x_i) \parallel \rho(f(x_i, l))))$$
where
$$KL(p \parallel q) = p \ln \left(\frac{p}{q}\right) + (1 - p) \ln \left(\frac{1 - p}{1 - q}\right)$$
is the Kullback-Leibler divergence (or binary relative entropy) between two probability distributions p and q. In our case, they correspond to the distribution from the prior model $P(Y_i[l] = 1|x_i)$ and to the distribution from the constructed model $\rho(f(x_i, l))$, where $\rho(x)$ is the logistic function $1/(1 + e^{-x})$. This term is basically the distance from the existing model to the new model built with newly labeled data. In the marginal case, if these two distributions are always the same, then the KL term will be zero and the loss function will be exactly the same as the first term, which is nothing but the logistic loss. Here, η is used to control the relative importance of these two terms. This weight may be determined empirically on a held-out set.

Note that most classifiers support a way of combining models or augmenting the existing model, so although this implementation is classifier (i.e. Boosting) dependent, the

idea is more general. For example, in a Naive Bayes classifier, this can be implemented as linear model interpolation or a Bayesian adaptation (like MAP) can be employed.

3.2. Combining Adaptation with Active Learning

As an extension of this adaptation method, we propose to combine it with active learning [13]. Active learning aims to minimize the number of labeled utterances by automatically selecting the utterances that are likely to be most informative for labeling. The idea is using the existing model to selectively sample the utterances to label for the target application, and do the adaptation using those utterances. This technique is supposed to eliminate the labeling of the examples or classes which are already covered by the existing model. It is especially important to determine the initial set of examples to label when the labeling resources are scarce.

Since there is a previous model to be used to get confidences for the examples from the target application, we employ certainty-based active learning [14]. In this algorithm, the existing model labels the unlabeled examples and determines the "certainty" or "confidence", $P(Y_i[l] = +1|x_i)$, of each of its predictions. The examples with the lowest certainty levels are then presented to the labelers for labeling.

4. EXPERIMENTS AND RESULTS

We have evaluated the proposed adaptation method using the utterances from the database of the VoiceTone®[1] system. We have selected two applications, T_1, and T_2, both from the telecommunications domain, where users have requests about their phone bills, calling plans, etc. The first one is a concierge-like application which has all the intents the second application covers. The second one is used only for a specific subset of intents. The data properties are shown in Table 1. As seen the perplexity (computed using the prior distributions of the intents) of the second application is significantly lower while the utterances are longer. We have about 9 times more data for the first application. All the data is transcribed. We have performed our tests using the Boostexter tool [1]. For all experiments, we have used word trigrams as features. In order not to deal with finding the optimal iteration numbers, we have iterated many times, got the error rate after each iteration and used the best error rate in all the results below.

In this experiment, the goal is adapting the classification model for T_1 using T_2 so that the resulting model for T_2 would perform better. Table 2 presents the baseline results using training and test data combinations. The rows indicate the training sets and columns indicate the test sets. The values are the classification error rates, which are the ratios of the utterances for which the classifier's top scoring class is not one of the correct intents. The third row is simply the concatenation of both training sets (indicated by $simple$). The fourth row (indicated by $tagged$) is obtained by training the classifier with an extra feature indicating the source of that utterance, either T_1 or T_2. The performance of the adaptation is shown in the last 3 rows (indicated by $adapt$). As seen, although the two applications are very similar, when the training set does not match the test set, the performance drops drastically. Adding T_1 training data to T_2 does not help, actually it hurts significantly. This negative effect disappears when we denote the source of the training data, but no improvement has been observed on the performance of the classification model for T_2. Adaptation experiments using different η values indicate interesting results. We have seen that using a value of 0.1, it is actually possible to outperform the model performance trained using only T_2 training data.

Since we expect the proposed adaptation method to work better with less application specific training data, we have drawn the learning curves as presented in Figure 2 using 0.1 as the η value. The top most curve is obtained using random selection of only T_2 training data. When we employ adaptation with only 1,106 utterances from T_2, we have seen 2.5% absolute improvement, which means 56% reduction (from about 2,500 utterances to 1,106 utterances for an error rate of 16.8%) in the amount of data needed to achieve that performance. Then we combine supervised adaptation with active learning where we selectively sample the training data using the previously trained model, and get a further boost of another 1% absolute, making the reduction in

	T_1	T_2
Training Data Size	53022	5866
Test Data Size	5529	614
Number of Intents	121	98
Call-Type Perplexity	39.4	14.7
Average Utterance Length	8.1	10.6

Table 1. Data characteristics used in the experiments.

Training Set	Test Set	
	T_1	T_2
T_1	14.4%	26.9%
T_2	36.4%	13.4%
$simple$	14.2%	16.8%
$tagged$	14.1%	13.4%
$adapt(\eta = 0.1)$	19.0%	12.5%
$adapt(\eta = 0.5)$	16.1%	14.0%
$adapt(\eta = 0.9)$	15.3%	16.0%

Table 2. Adaptation results for the experiments. "$simple$" indicates simple concatenation, "$tagged$" indicates using an extra feature denoting the source of training data, "$adapt$" indicates adaptation with different η values.

[1] VoiceTone® system is provided by AT&T for customer care centers.

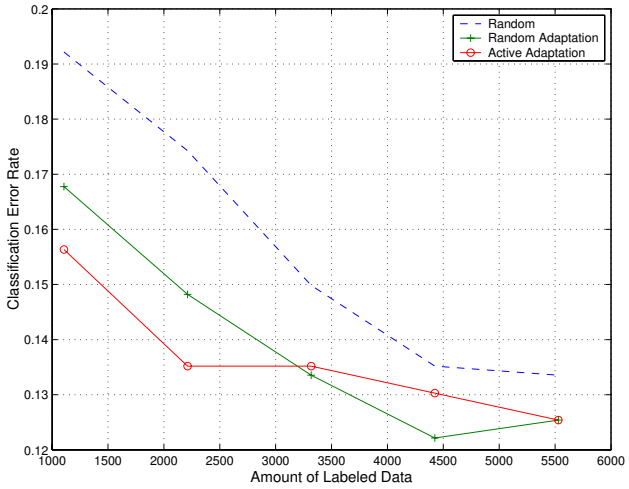

Fig. 2. Results using intent classification model adaptation. Top most learning curve is obtained using just T_2 data as a baseline. Below that lie the learning curves using the adaptation with random and selective sampling.

the amount of data needed 64% (from about 3,000 utterances to 1,106 utterances for an error rate of 15.6%.) Both adaptation curves meet at the end, since the pool where we select the utterances from T_2 is fixed already. One interesting point is that, after about 3,250 utterances, the adaptation with random sampling curve outperforms the adaptation with selective sampling curve. In a real-life scenario we can expect the adaptation with active learning curve to outperform random adaptation curve where the pool of candidate data is not fixed apriori.

5. CONCLUSIONS AND DISCUSSION

We have presented a supervised adaptation method for natural language intent classification using Boosting. We have shown that, for this task, using the proposed adaptation methods, it is possible to boost the performance of a spoken language understanding system when there is not much training data available. We have also proposed combining supervised adaptation with active learning. Our results indicate that we have achieved the same intent classification accuracy using around 60% less labeled data.

Although this implementation is classifier (namely, Boosting) dependent, the idea is more general. It is also possible to apply the idea for other classification tasks which may need adaptation such as topic classification or named entity extraction. The very same adaptation technique can be employed in cases where there is a mismatch between the training and the target application.

Our future work includes unsupervised adaptation of intent classification models. This will enable us to bootstrap new spoken dialog systems without labeling any application specific data.

Acknowledgments We would like to thank Dilek Hakkani-Tür, Murat Saraclar, Mazin Rahim, and Robert E. Schapire for many helpful discussions.

6. REFERENCES

[1] 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.

[2] P. Haffner, G. Tur, and J. Wright, "Optimizing SVMs for complex call classification," in *Proceedings of the ICASSP*, Hong Kong, April 2003.

[3] R. E. Schapire, "The boosting approach to machine learning: An overview," in *Proceedings of the MSRI Workshop on Nonlinear Estimation and Classification*, Berkeley, CA, March 2001.

[4] G. Riccardi and A. L. Gorin, "Stochastic language adaptation over time and state in a natural spoken dialog system," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, pp. 3–9, 2000.

[5] M. Bacchiani, B. Roark, and M. Saraclar, "Language model adaptation with MAP estimation and the perceptron algorithm," in *Proceedings of the HLT-NAACL*, Boston, MA, May 2004.

[6] V. Digalakis, D. Rtischev, and L. G. Neumeyer, "Speaker adaptation using constrained estimation of gaussian mixtures," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 5, 1995.

[7] Y. He and S. Young, "Robustness isuues in a data-driven spoken language understanding system," in *Proceedings of the HLT/NAACL Workshop on Spoken Language Understanding*, Boston, MA, May 2004.

[8] B. Roark and M. Bacchiani, "Supervised and unsupervised PCFG adaptation to novel domains," in *Proceedings of the HLT-NAACL*, Canada, May 2003.

[9] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, 1999.

[10] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *The Annals of Statistics*, vol. 38, no. 2, pp. 337–374, 2000.

[11] R. E. Schapire, M. Rochery, M. Rahim, and N. Gupta, "Incorporating prior knowledge into boosting," in *Proceedings of the ICML*, Sydney, Australia, July 2002.

[12] G. Tur and D. Hakkani-Tür, "Exploiting unlabeled utterances for spoken language understanding," in *Proceedings of the Eurospeech*, Geneva, Switzerland, September 2003.

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

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