Automatically Optimizing Utterance Classification Performance without Human in the Loop

Yun-Cheng Ju, Jasha Droppo

Speech Research Group, Microsoft Research, Redmond, WA. USA

{yuncj,jdroppo}@microsoft.com

Abstract

The Utterance Classification (UC) method has become a developer's choice over traditional Context Free Grammars (CFGs) for voice menus in telephony applications. This data driven method achieves higher accuracy and has great potential to utilize a huge amount of labeled training data. But, having a human manually label the training data can be expensive. This paper provides a robust recipe for training a UC system using inexpensive acoustic data with limited transcriptions or semantic labels. It also describes two new algorithms that use caller confirmation, which naturally occurred within a dialog, to generate pseudo semantic labels. Experimental results show that, after having sufficient labeled data to achieve a reasonable accuracy, both of our algorithms can use unlabeled data to achieve the same performance as a system trained with labeled data, while completely eliminating the need for human supervision.

Index Terms: Call Routing, Statistical grammars, Spoken language understanding (SLU), Utterance Classification (UC)

1. Introduction

Recently, Utterance Classification (UC) has been widely applied to the task of call routing [1, 2]. It is statistical and data-driven in nature and is typically used to classify natural spoken responses to open-ended prompts like "*How may I direct your call?*"

A main advantage of UC is that the performance of a deployed system can be continuously improved over time. Encouraging results have been reported where huge amount of logged utterances were transcribed, semantically labeled, and then used to adapt to both the language model and the classifier [3]. However, the most costly steps in this methodology, namely the transcription and semantic labeling, can only be partially automated and still require human supervision to some extent.

The first research challenge we address in this paper is to find a methodology to best utilize all of the data available, regardless of whether it is fully transcribed, semantically labeled only, or completely unlabeled. Fully transcribed utterances have historically been most useful and necessary to train UC systems. While expensive, they may also have been generated on a subset of the data as part of a contractual service agreement. Semantically labeled utterances do not have a transcription, and can be generated for a much smaller cost. Because call volume is generally much higher than the number of affordable transcribed or labeled utterances, the bulk of the utterances remain unlabeled.

In this paper, we demonstrate a system that learns from semantic labels instead of transcriptions with only a small loss in accuracy. We also introduce two new algorithms that use naturally occurring confirmation information from the dialog system to provide 'pseudo semantic labels' and completely eliminate the requirement of human supervision in the optimization loop.

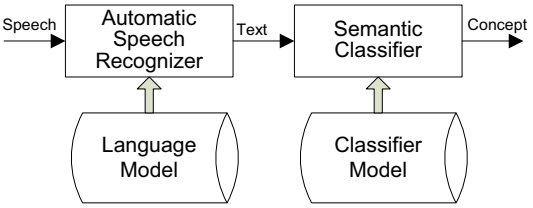

Figure 1: *Utterance Classification (UC) system diagram*

The remainder of this paper is organized as follows. The details of the proposed methods are described in Section 2. Experimental results are reported in Section 3 and Section 4 concludes the paper.

2. Methods

The standard UC process is depicted in Figure 1. The speech recognizer uses a language model, most likely statistical in nature, to transform the speech utterance into words. It is followed by a text classifier to further transform the recognized words into a fixed set of semantic concepts.

This procedure requires training of both the language and classifier models, as explained in the next subsections.

2.1. Statistical language model

In general, a better statistical language model (SLM) can help the speech recognizer reduce Word Error Rate (WER) in transcribing speech into text. Because research has shown that the configuration that achieves the lowest WER might not always give the lowest Classification Error Rate (CER) [2], our research focuses on building language models that produce consistent transcriptions, but not necessarily the lowest WER.

In addition, as manual transcription of speech utterances remains a major bottleneck in model tuning with large amount of training data, we are interested in semi-supervised learning algorithms which use some transcriptions in the initial phase but do not require transcriptions for the rest of the corpus.

Given the above goals, we adopted the iterative SLM self-adaption paradigm of [2]. We'll demonstrate that this approach not only achieves higher classification accuracy but also enables the use of more acoustic data without any transcriptions.

We use all of the available transcriptions to train an initial SLM. We then use this SLM and all the available acoustic data, transcribed or not, to generate ASR transcriptions to train a second SLM. We repeat this process for several iterations.

2.2. Classification model

A classification model serves as a routing table to map a text string into a fixed set of semantic concepts. Both Maximum Entropy [4] and Vector Space Model [5] based classifiers have been widely used in call routing applications. We use the exact classifier setup from our previous work [6] using bigram word features. In our formulation, the discriminant function

controls how the elements of the routing matrix R map the terms of the query vector $\boldsymbol{x}$ into a target class (voice menu option) j.

$$g_j(\boldsymbol{x}, R) = \langle \boldsymbol{x}, \boldsymbol{r}_j \rangle = \sum_k x_k r_{jk}$$

The routing matrix is initialized using the TF-IDF formula [7] and then discriminatively trained based on Minimum Classification Error (MCE) criterion using procedures similar to [1] to guarantee a minimized CER. Details of the parameters and training algorithms can be found in [1, 6].

We use the ASR transcriptions, described in the previous section, of the semantically labeled (i.e., acoustic data with a semantic label) subset of acoustic data to train the classifier.

2.3. Pseudo semantic labels from user confirmation

Logging provides important insight to the entire dialog activity, including dialog stage, system prompt, caller response, and the recognition context. A common practice in telephony applications is to use an additional confirmation turn such as "*I think you said **weather**, is that right?*" at each major branch point to guarantee the dialog is moving in the right direction. If the answer is a clear "*yes*", then the caller's *accepted confirmation* has semantically labeled the utterance for us. But, can we learn anything from the *denied confirmation* when the caller says "*no*"?

One use of the denied confirmation (e.g., "not weather") is to improve the automatically generated pseudo semantic labels during training. Because the label associated with the denied confirmation was the best label chosen by the deployed system, it is likely to be at the top of the classifier N-best candidate list during training. Because we know this label is incorrect, it can be safely removed and replaced with the classifier's second choice. In addition, this is a particularly important mechanism for some of the denied examples where the callers did not ask for a valid menu option but the deployed system failed to reject. Our algorithm can provide the classifier with training utterances that the application should not accept. It is not practical for the application to bring up any negative confirmation like "*I don't think I should pay attention to what you said, right*?"

We propose two algorithms (**relaxation** and **progression**) to produce the pseudo semantic labels for the denied examples. Both algorithms iteratively update the pseudo semantic labels from the improved classification results, expecting the new pseudo labels be more accurate and lead to an even more accurate classifier for the next iteration. We believe that more correct semantic labels will produce lower CER.

The *relaxation* algorithm is shown in Figure 2. Starting with a baseline classifier configuration (e.g., with only the accepted confirmations and human labeled utterances), it iteratively re-generates a more trustworthy set of pseudo semantic labels for all denied confirmations using the hopefully more and more accurate classification results. The semantic labels of the accepted confirmations and labeled utterances are not changed. It repeats until the pseudo labels stabilize. Notice the pseudo semantic labels for the entire set of the denied confirmation utterances are updated at every iteration.

In contrast, the *progression* algorithm gradually assigns the pseudo semantic labels in smaller groups throughout the span of the optimization. In each iteration, the entire set of denied confirmations receive new pseudo labels using the most current classifier result and the *unsettled* ones (to be explained) are randomly partitioned into small groups. Each

Initialize the baseline classifier
Loop:
  Assign pseudo semantic labels for the denied confirmations using the current classifier output
  Retrain the classifier
  Repeat the Loop if any of the pseudo labels changes

Figure 2: *Relaxation Algorithm for Pseudo Labels*

Initialize the baseline classifier
Outer Loop:
  Update pseudo semantic labels for the denied confirmations using the current classifier output
  Randomly partition the un-settled examples of the denied confirmations into smaller groups
  Inner Loop: For each of the smaller groups
    Train a classifier using the current training data and the group.
    Evaluate the heuristic discrepancy score over all examples.
  Select the most promising configuration for this iteration
  Settle the small group of newly updated pseudo labels
  Add the examples into the current training set.
  Repeat the Outer Loop until all pseudo labels are settled

Figure 3: *Progression Algorithm for Pseudo Labels*

small group of the unsettled denied confirmations, with new pseudo labels, is separately combined with the current training data to form a candidate configuration. Finally, each new candidate is evaluated against the entire training set to count the number of *discrepancies*. A discrepancy is when the classifier disagrees with an accepted confirmation or a human labeled utterance, or when it produces a confirmation that has been denied. Even though the correct semantic label remains a mystery, the discrepancy count serves as a heuristic function to evaluate the potential performance of the new classifier configuration. We select the candidate configuration with the lowest number of discrepancies, settle the pseudo semantic labels for its small group of denied examples, and repeat the outer loop. Once a pseudo label is settled, even though it is still used the same way in the discrepancy calculation like other denied confirmations, the new label stays fixed and is used for subsequent training iterations. The algorithm stops when all denied examples are settled.

This approach is certainly computationally more expensive. However, re-training the classifiers is fast and the inner loop can be completely executed in parallel. The algorithm is illustrated in Figure 3.

3. Experiments

Experiments were performed using phone calls collected from the main-menu call routing task in Ford Sync online services [8], consisting of a total of 20,216 fully transcribed and semantically labeled utterances corresponding to the system prompt "*Services. Which service do you want?*" These calls were collected throughout the first 9 months of the service deployment. There are 10 menu options including "Business Search", "Driving Directions", "Traffic", "Weather", "Sports", "Favorites", "News", "Next Turn", "Tell Me my Choices", and "Good bye". Roughly 6.5% of the utterances which are entirely noise, side speech, or completely irrelevant to the task (e.g., "Hello", "Bring me a woman") were assigned a unique 'Reject' category. All categories were treated equally in the CER calculation.

Utterances were randomly divided into the training set of 12,216 utterances and the test set of 8,000 utterances. Furthermore, in order to study the impact of the corpus size on the performance, we created several configurations in which only a fraction (down to only 1/16) of the training set were used.

For all of the experiments performed, we used the same acoustic model and decoder setup as reported in [6].

3.1. Using data without transcriptions or labels

To first evaluate our iterative language model training and the accommodation of un-transcribed audio data, we experimented with 3 different SLM building options. The classification accuracies are illustrated in Figure 4. The x-axis shows the different number of the fully transcribed and semantically labeled initial SLM and classifier training sentences. Notice the different classification performance among the 3 curves is attributed entirely to the different ASR transcriptions.

The baseline (option 'Human Transcription') was trained directly with clean transcriptions of the training set (with various sizes). As expected, the performance improves with more transcribed and semantically labeled training sentences. We then iteratively trained the SLM as described in Section 2.1 for another 4 iterations with the same transcribed audio data (option 'Self-Adaptation'). The iteratively trained SLM from ASR output generally outperforms the SLM trained from only human transcriptions.

Finally, for the option 'Plus Un-transcribed Audio', we restarted the iterations from the same baseline SLM also for 4 iterations but with all of the audio data available. In the previous two options, only the last configuration (x=12,216) uses all of the 12,216 audio utterances. Clearly, this option gains additional accuracy from using the rest of the training set as un-transcribed audio data.

Next, with the best SLMs trained from the previous experiment, we examined the value of human semantic labels to un-transcribed acoustic data. The green curve (option 'No Additional') in Figure 5 is equivalent to the best SLM training configuration, with no additional semantically labeled data (the 'Plus Un-transcribed audio' option in Figure 4). Each additional curve in Figure 5 shows the classification accuracy achieved with a specific number of classifier training sentences. Notice that the number of observation data points decreases as more semantically labeled data was used in the baseline. In addition, when the semantic labels of all of the data are available as shown in the top curve (12,216), the different performance is attributed entirely to the quality of the initial SLM, which is dominated by the manual transcriptions available.

As shown in Figure 5, performance might not always improve when we add a small amount of additional training data. However, having all of the data semantically labeled achieves the best performance, no matter how many transcribed utterances we had when building our initial SLM.

The results also suggest that, when the quality of the initial SLM is low, the best way to improve the classification performance is to fully transcribe more utterances. But, transcriptions become less effective as the number of the transcribed utterances increases. Starting at the earlier configuration (accuracy = 93.44%, x=764), fully transcribing and labeling another 763 (1,527-764) utterances quickly improves the accuracy to 94.73%. However, labeling another 5,344 (6,108-764) utterances alone only improves the accuracy to 94.64%. On the other hand, starting at the later configuration (accuracy = 95.3%, x=6,108), as long as we have the semantic labels of the remaining 6,108 (12,216-6,108) utterances, there is almost no difference in accuracy (both are around 95.7%) whether we fully transcribe those utterances or not.

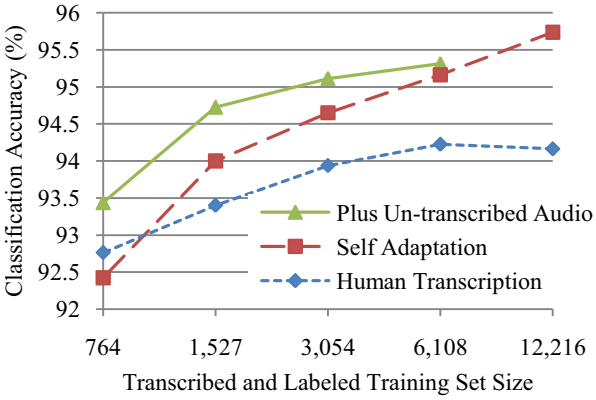

Figure 4: *Accuracy for different SLM Training Options*

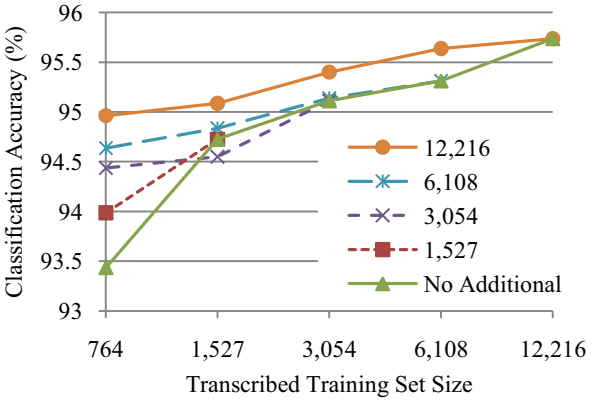

Figure 5: *Performance Improvement with additional Semantically Labeled but Un-transcribed Classifier Training data*

3.2. Tuning without human in the loop

We investigate the feasibility of performance tuning using just the caller confirmation information from the logged calls without any human supervision and compare the performance of the two new algorithms we proposed.

We use the best configurations, corresponding to 'Self Adaptation' in Figure 4, as reference systems. To simulate the behavior of a real application, utterances classified as 'Reject' are automatically discarded (i.e., no opportunity for caller confirmation). In addition, to simplify the study, we also assume the caller faithfully accepted or denied all of the confirmations and the system correctly recognized them. For example, we marked a case as 'not Traffic' if the caller said something that should have been rejected but was mis-classified as 'Traffic'. We have a total of 12,216 fully semantically labeled training utterances; therefore configurations built with more training sentences have a smaller amount of simulated logged data.

Figure 6 illustrates our feasibility study. We randomly selected and added 80% of the additional utterances from the labeled training corpus to each configuration, and found almost no difference in performance compared with having the correct semantic labels for the entire data set. Since the reference system is operated at the accuracy of 93% and higher, one might expect this to be the performance we can reliably gain from logged data. Therefore, we examined a

naïve approach which trusts the classifier's rejection by labeling all rejected utterance as 'Reject' and discards all of the utterances the caller denied (i.e., without using the examples of denied confirmation at all). There was almost no performance improvement in the naïve approach, mainly because it didn't add anything which the system didn't already know.

In addition, not having those rejected utterances limits the improvement in the classification performance. The '*Upper Bound*' curve in Figure 6 shows the ideal performance of using caller confirmations even if both 1) the correct semantic labels for all of the denied examples are known; and 2) the correct distribution for the 'Reject' class in the new training set is properly maintained.

Having these artificial systems to compare with, we examine the performance of our two new algorithms which produce pseudo semantic labels from the call logs to automatically improve the system performance without human supervision.

As illustrated in Figure 7, the *progression* approach slightly outperforms the *relaxation* approach. Both approaches reach the *Upper Bound* performance in the configurations where the deployed system is more accurate.

For the '*Oracle*' configuration, only the correct pseudo semantic labels are included to train the new classifier. Even for the configurations where the starting classifiers are less accurate, the progression algorithm achieved more than 88% of the relative performance improvement only the *Oracle* can deliver.

If it is practical for humans to transcribe or semantically label the entire training set, the unsupervised methods provide less value. However, when the system has already been trained from a large amount of data and the accuracy is good, existing methods would need an even larger amount of semantically labeled data to marginally further improve its performance, which is prohibitively costly. Our algorithms are most useful in this case because the same performance gain can be achieved with no human supervision.

We expect both algorithms to perform even better as the baseline system keeps improving and more logged utterances become available.

4. Conclusions

While transcriptions and semantic labels are still useful and provide the best performance, lack of them is no longer an obstacle in our proposed approach. Utilizing the caller confirmation and the rich classification N-best candidates, our new algorithms can use a large amount of logged training data to automatically improve its performance completely without any human supervision in the loop. Our approach can take advantage of all available data: the transcribed utterances can be used in initial SLM training, while all semantically labeled data can be used in classifier training. Even the un-labeled, un-transcribed audio data is useful because we use the ASR transcriptions of all audio data to iteratively train the SLM to further improve classification accuracy.

Future work includes applying active learning methods to select the best subset of the rejected utterances and the denied confirmation samples for human semantic labeling; and the investigation on the effect and use of noisy confirmation data due to user and system errors.

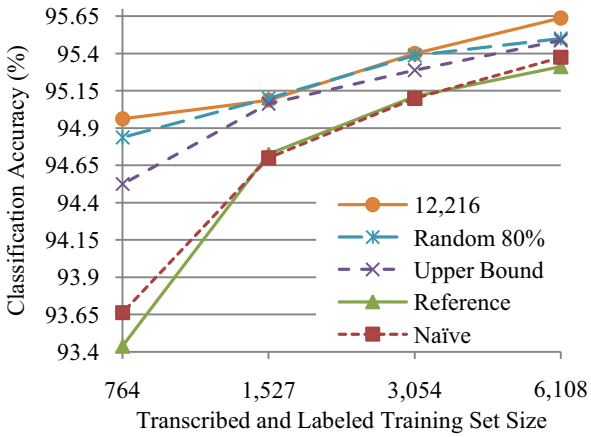

Figure 6: *Feasibility Study of Tuning without Human Supervision*

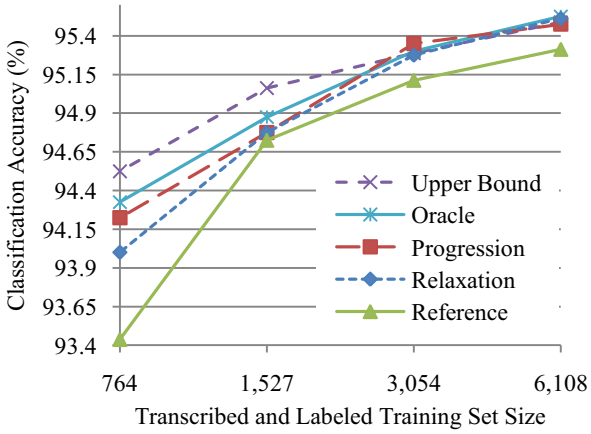

Figure 7: *Performance Improvement without Human Supervision*

5. Acknowledgements

The authors would like to thank Issac Alphonso, Xiaoqiang Xiao and Geoffrey Zweig for many useful discussions in this project.

6. References

[1] Hong-Kwang Jeff. Kuo and Chin-Hui Lee, "Discriminative training in natural language call routing," in Proc. of ICSLP, Beijing, China, 2000.

[2] Ye-Yi Wang, J. Lee, and A. Acero, "Speech utterance classification model training without manual transcriptions," in Proc. of IEEE ICASSP, Toulouse, France, 2006.

[3] D. Suendermann, K. Evanini, J. Liscombe, P. Hunter, K. Dayanidhi, R. Pieraccini, "From rule-based to statistical grammars: continous improvement of large-scale spoken dialog systems", in Proc. of IEEE ICASSP, 2009.

[4] A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra, "A maximum entropy approach to natural language processing." Computational Linguistics, vol. 22, 1996.

[5] J. Chu-Carroll and B. Carpenter, "Vector-based natural language call routing," Computational Linguistics, vol. 25, no. 3, 1999.

[6] Xiaoqiang Xiao, Jasha Droppo, and Alex Acero, "Information retrieval methods for automatic speech recognition", in Proc. of IEEE ICASSP, 2010.

[7] G. Solton and C. Buckley, "Term weighting approaches in automatic text retrieval", Information Processing and Management, 24(5):513-523, 1988.

[8] http://www.fordvehicles.com/technology/sync