EXTENDING BOOSTING FOR CALL CLASSIFICATION USING WORD CONFUSION NETWORKS

Gokhan Tur, Dilek Hakkani-Tür, Giuseppe Riccardi

AT&T Labs-Research,
180 Park Avenue, Florham Park, NJ, USA
{gtur,dtur,dsp3}@research.att.com

ABSTRACT

We are interested in the problem of robust understanding from noisy spontaneous speech input. In goal driven human-machine dialog, utterance classification is a key component of the understanding process to determine the intent of the speaker. In this paper we propose a novel algorithm for exploiting ASR word confidence scores for better utterance classification of spoken utterances. Word confidence scores for automatic speech recognition (ASR) provide estimates for word error rates. While previous work has focused on straightforward combination of word confidence scores into Bayesian classifiers, in this paper we extend the mathematical formulation for Boosting classifiers. This extension of the algorithm allows to exploit confidence scores from a 1-best ASR output or from *word confusion networks* (WCNs). We present methods for on-line and off-line score combinations. The results we show are for a large database of utterances collected using the AT&T VoiceTone^{SM} spoken dialog system. Our experiments show between 5%-10% reduction in error ($1 - precision$) for a given $recall$ using WCNs compared to ASR output.

1. INTRODUCTION

In goal driven human-machine dialog, utterance classification is a key component of the understanding process to determine the intent of the speaker, i.e. *call-type*. Both automatic speech recognition and utterance classification decoding are noisy and in general their noise statistics are *not* correlated. Our goal is to exploit the error estimates from ASR to improve the user intent classification.

Word confidence scores for ASR provide estimates for word error rates. ASR confidence scores have been used for improved rejection of noisy or not *well formed* utterances [1, among others]. More generally, we can expect to process a set of ASR hypotheses with confidence scores, i.e word lattices (WL). Word lattices encode a very large set of word strings and allow for a tighter integration of the ASR search space and the classification process. A special case of word lattices are the word confusion networks [2, 3]. WCNs carry the multi-string alignment information which is missing from WL alone, have better word confidence scores, and are more compact from a memory/process-time point of view [3].

In previous work we have presented methods to exploit word confidence scores [4, 5] and WCNs [5] for a Bayesian classifier with feature selection. In this paper we propose a novel algorithm for exploiting ASR word confidence scores with an extended mathematical formulation for Boosting family of classifiers. This extension allows to exploit confidence scores from a 1-best ASR output or from WCNs.

In the next section, we briefly present our algorithm to compute WCNs from WLs. After presenting the basics of the Boosting classifier we have employed, we address how we use WCNs for better classification. Section 5 includes the experiments we have conducted.

2. WORD CONFUSION NETWORKS

Word confusion networks provide a very compact representation of multiple ASR hypotheses, preserving the richness and accuracy of the word lattices. Typical structures of WCNs compared to WLs is given in Figure 1. A detailed explanation of our algorithm for composing WCNs from WLs and the comparison of its performance with other approaches is presented in [3]. A summary of this algorithm is as follows:

1. Compute the posterior probabilities for all transitions in the lattice.
2. Extract a path from the lattice (which can be the best, longest or a random path), and call this as the *pivot* of the alignment.
3. Traverse the lattice, and align all the transitions with the pivot, merging the transitions that correspond to the same word (or label) and occur in the same time interval (by summing their posterior probabilities).

The output of this algorithm is a more compact structure, called WCN.

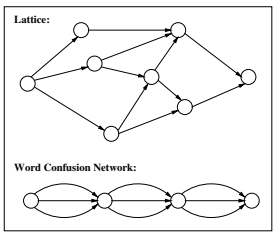

Fig. 1. Typical structures of WLs and WCNs.

3. BOOSTING

We begin by a review of Boosting-style algorithms. Boosting aims to combine "weak" base classifiers to come up with a "strong" classifier [6]. This is an iterative algorithm, and in each iteration, a weak classifier is learned so as to minimize the training error.

The algorithm generalized for multi-class and multi-label classification is given in Figure 2. 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 $l \in \mathcal{Y}$ to be

$$Y[l] = \begin{cases} +1 \; if \; l \in Y \\ -1 \; if \; l \notin Y \end{cases}$$

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 combination which is easier to classify gets lower weights and vice versa. The intended effect is to force the weak learning algorithm to concentrate on examples and labels that will be the most beneficial to the overall goal of finding a highly accurate classification rule. The final strong "strong" learner, H, is then nothing but a linear combination of the individual weak learners, h_t.

4. EXPLOITING WORD CONFUSION NETWORKS

We propose various methods to exploit WCNs. In all of them the main idea is to extend the feature space of the classification algorithm. The first method is independent of the classifier used. It aims to filter or bin the features used according to the word confidences obtained from the WCN. The second and third ones are specific to Boosting. One of them aims to change the weak learner found, and the other to change the way weak learners are used during run-time.

4.1. ASR Output Filtering and Binning

The simple approach for handling ASR errors during classification is filtering words that have very low confidence scores from the ASR output during run-time. In the cases where the ASR output for the training data is also available, binning the training and test words using their confidence scores is another baseline approach. To implement binning, each word w in the training data is replaced with w_0 if its confidence score is less than or equal to a threshold t_0, and w_i if its confidence score is in $(t_{i-1}, t_i]$, where $t_0 < t_1 < ... < t_{i-1} < t_i < ... < t_k = 1$.

- Given the training data from the instance space X: $(x_1, Y_1), ..., (x_m, Y_m)$ where $x_i \in \mathcal{X}$ and $Y_i \subseteq \mathcal{Y}$
- Initialize the distribution $D_1(i,l) = \frac{1}{mk}$
- 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.
- Then the output of the final classifier is defined as:

$$H(x,l) = sign(f(x,l))$$

where

$$f(x,l) = \sum_{t=1}^{T} \alpha_t h_t(x,l)$$

Fig. 2. The algorithm *Adaboost.MH*.

4.2. Extended Boosting Algorithm

Schapire and Singer [6] have proved a bound on the training error rate, i.e. *Hamming Loss* (HL), of H in the Boosting algorithm.

$$HL(H) \leq \prod_{t=1}^{T} Z_t \quad (1)$$

where Z_t is the normalization factor computed on round t:

$$Z_t = \sum_{l \in \mathcal{Y}} \sum_{i=1}^{m} D_t(i,l) e^{-\alpha_t Y_i[l] h_t(x_i,l)}$$

and HL is defined as the fraction of misclassified examples, i, and labels, l:

$$HL(H) = \frac{1}{mk} \sum_{l} \sum_{i} \delta(x_i, l)$$

where

$$\delta(x_i,l) = \begin{cases} 1 \; if \; H(x_i,l) \neq Y_i[l] \\ 0 \; otherwise \end{cases}$$

The bound in Equation 1 is important, because in order to minimize this training error, it is a reasonable approach to minimize Z_t on each round of Boosting. This leads to criteria for finding weak hypotheses, $h(x,l)$ for a given iteration. Assume that weak learners, h, make their predictions

based on a partitioning of the domain $\mathcal{X}$ into disjoint blocks X_j. Let $c_{jl} = h(x, l)$ for $x \in X_j$, then Schapire and Singer have proved that the optimal c_{jl} is given by:

$$c_{jl} = \frac{1}{2} ln \left(\frac{W_+^{jl}}{W_-^{jl}} \right)$$

where

$$W_b^{jl} = \sum_{i:x_i \in X_j} D(i,l)(1 - \delta(x_i, l))$$

Following this terminology, one can define the output of the Boosting classifier for each class for a given utterance, $x \in X_j$, as follows:

$$f(x,l) = \sum_{t=1}^{T} \alpha_t \times c_{jl}^t$$

where c_{jl}^t is the weak learner c_{jl} for iteration t. In this work we employed Boostexter [7] tool and used word n-grams as features. In that framework, the weak learners partition the domain with respect to the absence or presence of a feature. For call classification, where the input of the classifier is the ASR output, it is clear that instead of making the binary decision of absence or presence of a feature, it makes more sense to exploit confidences obtained from the word confusion networks. This requires the following extension to the above formula:

$$f(x,l) = \sum_{t=1}^{T} \alpha_t \times p(j) \times c_{jl}^t$$

where p(j) is the probability of being in the partition X_j, estimated by the confidence scores of the WCN. Note that this change takes place only during run-time. We assume that the weak learners are trained from transcribed data, hence does not contain any confidence.

4.3. Boostexter with Scored Features

Boostexter has already the capability of exploiting the scores associated to the features. These scores can be any real number and is not limited to $[0 - 1]$. Instead of using two partitions, (i.e. absence and presence of a feature), the weak learner tries to come up with three partitions, i.e. absence, presence with a score more (and less) than some threshold. This threshold is also learned from training data automatically so as to optimize the individual weak learner. Unlike the previous approach, this method requires the confidences for the training data is available and changes the model learned.

5. EXPERIMENTS AND RESULTS

In order to evaluate the proposed methods to exploit WCNs, we have used the utterances collected from an AT&T VoiceTone^{SM} spoken dialog application. In this application, the users of the system are greeted by the open ended prompt of *How May I Help You?*. The system tries to understand the responses and act upon them. We first describe our data and the evaluation metrics used to compare model performances, then show results.

Training Data Size	9094
Test Data Size	5171
Number of Call-Types	84
Call-Type Perplexity	32.64
Average Length	10.66

Table 1. Data characteristics used in the experiments.

5.1. Data

Table 1 summarizes the amount of data used for training and testing for this application along with the total number of call-types, average utterance length, and call-type perplexity. Perplexity is computed using the prior distribution of the call-types in the training data.

5.2. Evaluation Metrics

Inspired by the information retrieval community, while evaluating the classification performance, we used mainly *recall* and *precision* allowing multiple call-types. Recall is defined as the proportion of all the true call-types that are correctly detected by the classifier. It is obtained by dividing the number of true positives by the sum of true positives and false negatives. Precision is defined as the proportion of all the accepted call-types that are also true. It is obtained by dividing true positives by the sum of true positives and false positives. True (False) positives are the number of call-types for an utterance for which the detected call-type has got a confidence above a given threshold, hence accepted, and is (not) among the correct call-types. False negatives are the number of call-types for an utterance for which the detected call-type has got a confidence less than a threshold, hence rejected, but is among the true call-types.

5.3. Results

The recall vs. precision numbers when we used the transcriptions of utterances during training, and the ASR output, as well as filtered ASR output of the test data is given in Figure 3. To select the threshold for filtering words from the ASR output of the test set, we computed confidence scores for the words in the training set. We divided all the words into bins using their scores. We selected the confidence score for which the percentage of misrecognized words is the same as the correctly recognized words. As a result, 0.63 was selected as the threshold, and 11.4% of

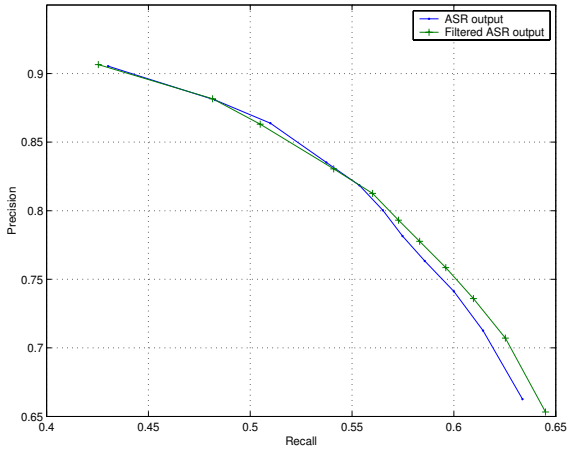

Fig. 3. Recall versus Precision when low confidence words are filtered from ASR output.

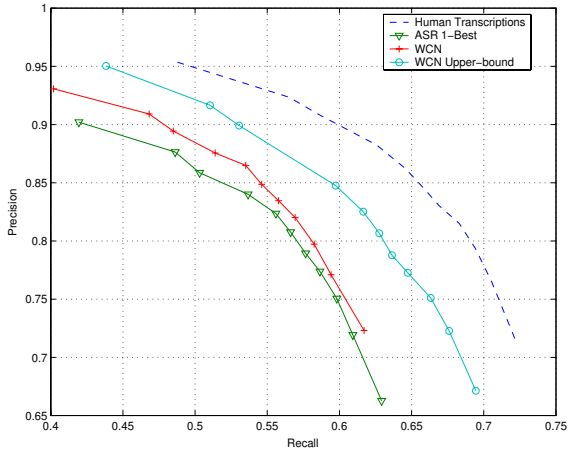

Fig. 4. Results obtained by extending Boosting with feature confidences.

the words are filtered from ASR output. Filtering low confidence words resulted in slightly better Precision at high Recall levels (see Figure 3). We also tried other thresholds, but could not obtain better results. Binning the words using their confidence scores did not result in significant improvements.

Figure 4 presents our results using extended Boosting. Top-most curve is obtained using human transcriptions and bottom-most one using ASR 1-Best using traditional Boosting. In order to show the potential improvement using WCNs, we have first made an experiment by keeping all the n-grams of the WCN occurring also in the human transcriptions. This is the upper bound we can get by using WCNs, shown by the curve with circles. The improvement using extended Boosting is shown by the curve just below that. As seen for all thresholds, we obtained better recall and precision values than using ASR 1-Best, and for larger thresholds we halved the way to the upper bound. On the average, for a given recall, we decreased the error rate $(1 - precision)$ by 5%-10%.

Using Boostexter with scored features did not help, actually we have got significantly inferior results on the test set although training set error rate was lower. This may be due to lack of training data to generalize or determine the threshold for each weak learner.

6. CONCLUSIONS

In this paper we have shown how to exploit the error estimates from ASR to provide better utterance classification. We presented a novel algorithm in the context of Boosting classifiers. The mathematical formulation of the algorithm allows to exploit confidence scores from a 1-best ASR output or from WCNs. We show between 5%-10% relative reduction in error rate using WCN compared to ASR output for a large database of utterances collected using the AT&T VoiceTone^{SM} spoken dialog system for customer care.

7. ACKNOWLEDGMENTS

We would like to thank Rob Schapire and Mazin Rahim for many useful discussions.

8. REFERENCES

[1] T. J. Hazen, S. Seneff, and J. Polifroni, "Recognition confidence scoring and its use in speech understanding systems," *Computer Speech and Language*, , no. 16, pp. 49–67, 2002.

[2] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech and Language*, vol. 14, no. 4, pp. 373–400, 2000.

[3] D. Hakkani-Tür and G. Riccardi, "A general algorithm for word graph matrix decomposition," in *Proceedings of the ICASSP*, Hong Kong, May 2003.

[4] R. C. Rose, H. Yao, G. Riccardi, and J. H. Wright, "Integration of utterance verification with statistical language modeling and spoken language understanding," *Speech Communication*, vol. 34, pp. 321–331, 2001.

[5] 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*, Denver, CO, September 2002.

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

[7] 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.