A DATA SELECTION STRATEGY FOR UTTERANCE VERIFICATION IN CONTINUOUS SPEECH RECOGNITION

Hui Jiang, Frank Soong and Chin-Hui Lee

Dialogue Systems Research, Multimedia Communication Research Lab, Bell Labs, Lucent Technologies, Murray Hill, NJ 07974, USA
Email: {hui,fks,chl}@research.bell-labs.com

ABSTRACT

In this paper, we propose the concept of rival for verifying hypothesis in speech recognition. A likelihood ratio test, based on the rivals model, are investigated for utterance verification in continuous speech recognition. We present a data selection strategy to identify useful subsets of training data to train rival model automatically from training data. And a single pass strategy for utterance verification, namely verification-in-search, is also proposed. Some preliminary experiments on DARPA Communicator travel task have shown the rival models give better verification performance in terms of identifying mis-recognized words from the output of our baseline recognizer.

1. INTRODUCTION

Recent advances in automatic speech recognition (ASR) technology have enabled ASR systems to migrate from laboratory to many services and products. However, in many practical applications, it becomes more desirable and urgent to equip a speech recognizer with utterance verification (UV).[4] Utterance verification is a procedure used to verify how reliable are the results from a speech recognizer. Usually, a quantitative score, also called confidence measure, is used to indicate the reliability of every recognition decision. Based on the confidence measure, a series of further actions can be taken after recognition, e.g., to reject or remedy the recognition results. Utterance verification is a crucial technique to make today’s speech recognizers more “intelligent” than ever before. For instance, a speech recognizer with a powerful UV capability will be able to smartly reject non-speech noises, detect/reject out-of-vocabulary words, even correct some potential recognition mistakes, guide the system to perform unsupervised learning, and provide side information to assist high level speech understanding, etc.

Extensive studies on utterance verification have been performed recently in the literature. One of the most important progresses is to cast utterance verification scenario as a statistical hypothesis testing problem.[4, 7] According to the Neyman-Pearson Lemma, an optimal test is to evaluate a likelihood ratio between two hypotheses, $H_0$ and $H_1$. However, the alternative hypothesis $H_1$ is a composite one and it consists of many heterogeneous events so that it is always very difficult to model $H_1$ appropriately in UV. In [4, 7], the same HMM model structure is adopted to model $H_1$, they are commonly named as anti-models. Some limited successes have been obtained in using anti-models to model the alternative hypothesis $H_1$ when anti-models are trained from some discriminative training procedure. However, we are still in search of a more powerful method to model this complicated hypothesis.

In this paper, we study the problem of utterance verification for large-vocabulary continuous speech recognition using the same hypothesis testing paradigm. We propose a concept of rival and suggest to use a rival model to perform utterance verification in continuous speech recognition. One effective rival model training procedure is proposed. Experimental results on the DARPA communicator task show that the performance of utterance verification based on mono-phone rival models is significantly better than that of the standard monophone anti-models.

2. UTTERANCE VERIFICATION BASED ON RIVAL MODEL

For a typical pattern classifier, given an observation $X$ as input, we always get a pattern class $W$ as its output. However, if we look at the input $X$ in more details, $X$ could come from several different sources: i) $X$ actually comes from the class $W$, i.e., a correct classification; ii) $X$ comes from other classes instead of $W$, i.e., a classification error; or iii) $X$ is an outlier, i.e., $X$ comes from none of classes registered in the classifier. Therefore, if an observation $X$ is classified as $W$ but it actually does not belong to the class $W$, we simply call it as a rival of the class $W$.

In speech recognition, we encounter the same situation. Given a speech utterance $X$ as input, a speech recognizer usually gives a linguistic unit $W$ as output. However, the input $X$ itself could be from the class $W$ or just a rival of $W$. A classical speech recognizer does not provide us too much information about whether $X$ is from $W$ or its rival.

In a speech recognizer, if it is based on the optimal Bayes decision rule, we can define the set of all rivals of $W$ as $S_r(W)$:

$$S_r(W) = \{ X \mid \Pr(W|X) > \Pr(W'|X), \forall W' \neq W, X \not\in W, \text{ and } \Pr(W|X) \geq \xi \}.$$  

(1)

where $X \not\in W$ denotes $X$ is not from class $W$ and $\xi > 0$ is a constant. In the above definition, any rival with too small a probability or likelihood ($< \xi$) are excluded because it is rather easy to detect them in most cases.

On the other hand, the set of all observations from $W$ is denoted as $S_o(W)$:

$$S_o(W) = \{ X \mid \Pr(W|X) > \Pr(W'|X), \forall W' \neq W \text{ and } X \not\in W \}.$$  

(2)

$^1W$ may be a phone, a syllable, a word, a phrase or a sentence.
where \( X \supseteq W \) stands for \( X \) is from class \( W \).

Given an observation \( X \) from either \( \mathcal{S}(W) \) or \( \mathcal{S}(\neg W) \) as input, the speech recognizer will always give \( W \) as its recognized result. The Bayes decision procedure in the recognizer usually is unable to present enough information on whether \( X \) belongs to \( \mathcal{S}(W) \) or \( \mathcal{S}(\neg W) \). Obviously, the capability of utterance verification totally depends on how well we can distinguish \( \mathcal{S}(W) \) from \( \mathcal{S}(\neg W) \). In this paper, statistical hypothesis testing is still adopted as a tool to separate \( \mathcal{S}(W) \) from \( \mathcal{S}(\neg W) \) statistically. Given a recognition result \( W \) from recognizer for the observation \( X \), in order to reject or accept \( W \), we test the null hypothesis:

\[
\mathcal{H}_0: \quad X \in \mathcal{S}(W)
\]

against the alternative hypothesis:

\[
\mathcal{H}_1: \quad X \in \mathcal{S}(\neg W).
\]

Comparing with the previous works on hypothesis testing in [4, 7], both the null hypothesis \( \mathcal{H}_0 \) and the alternative hypothesis \( \mathcal{H}_1 \) in our proposed study are well-defined from available data, which in turn will make our modeling problem easier. The simplest way to model \( \mathcal{S}(W) \) and \( \mathcal{S}(\neg W) \) is that we estimate two different models \( \lambda_+ \) and \( \lambda_- \), and \( \lambda_+ \) for \( \mathcal{S}(W) \) and \( \mathcal{S}(\neg W) \), respectively, based on all possible training data from each of the sets. In this paper, \( \lambda_+ \) is named as \( \text{rival model} \). For simplicity, we also call \( \lambda_- \) \( \text{positive model} \) and \( \lambda_+ \) \( \text{negative model} \). These models can be estimated from training data according to different criteria, such as maximum likelihood estimation (MLE) or minimum verification error (MVE) [5, 2]. Both positive model \( \lambda_+ \) and negative (rival) model \( \lambda_- \) can be different from the model used in recognition. In this work, we choose MHM for both \( \lambda_+ \) and \( \lambda_- \) at subword unit level, e.g., phones.

Once \( \lambda_+ \) and \( \lambda_- \) are given, utterance verification is operated as the following likelihood ratio test:

\[
\eta = \frac{p(X | \mathcal{H}_0)}{p(X | \mathcal{H}_1)} = \frac{\frac{p(X)}{p(X | \lambda_+)} \cdot \frac{p(X | \lambda_+)}{p(X | \lambda_-)} \cdot \frac{p(\lambda_+)}{p(\lambda_-)}}{p(X | \mathcal{H}_1)} \geq \tau
\]

where \( \tau \) is the decision threshold.

Many other studies have proposed to use the “competitors” information in recognition procedure to improve utterance verification, such as, N-Best in [6] and word-graph in [8]. None of these approaches uses any information in training data for verification. Our proposed method attempts to extract extra information from the available data by the concept of \( \text{rival} \). This information is usually ignored by the conventional recognizer’s training procedure. Another close work is the so-called \( \text{cohort} \) model [4], which is trained based on some pre-defined \( \text{cohort} \) sets, which are usually independent from the recognizer. On the other hand, in our method, the \( \text{rival} \) sets are completely defined by the recognizer.

### 3. Training Rival Model

In isolated word speech recognition, given a recognizer and a training database, it is straightforward to define \( \mathcal{S}(W) \) and \( \mathcal{S}(\neg W) \) for every isolated word \( W \). Then, \( \lambda_+ \) and \( \lambda_- \) can be easily estimated from all training data assigned to the corresponding set. However, in continuous speech recognition, it is more difficult to define the \( \text{rival} \) set than the isolated word case. For example, in large-vocabulary, continuous speech recognition, it is very hard to associate a definite part of data to the rival set because numerous boundaries are possible and they are all considered during the Viterbi decoding. As a result, any possible segmentation in an utterance could potentially become a rival. Obviously, an exhaustive search is too expensive to be affordable. In this paper, we propose an efficient way to identify \( \text{rivals} \) for different speech units in continuous speech recognition. Every utterance in training set is processed by Viterbi search algorithm. During search, all possible segments in all active paths are examined to identify \( \mathcal{S}(a) \) and \( \mathcal{S}(\neg a) \) for every phone \( a \).

Given a speech recognizer and a training database, at first, for every utterance, we generate the true phone segmentation by aligning the utterance with the given transcriptions. Next, we perform beam based Viterbi search for every utterance \( X \) in the database. During the search, at every time instant \( t \), we back-trace every word-ending active partial path and compare all phone tokens with the “true” phone segmentation to determine each particular segment should be assigned to \( \mathcal{S}(a) \) or \( \mathcal{S}(\neg a) \). The above procedure is carried out through all training data to collect \( \mathcal{S}(a) \) and \( \mathcal{S}(\neg a) \) for every phone \( a \). We denote the \( t \)-th active word-ending path at the time instant \( t \) as \( L_t(t) \). We back-trace \( L_t(t) \) to get all its word segments \( L(t) = \{ W_0, W_1, \ldots, W_k \} \), where \( W_k \) denotes the \( k \)-th previous word in the path. In our current implementation, we only back-trace the last word \( W_k \) to get all its phone segments. Assume that the previous \( W_i \) in the partial path \( L(t) \) consists of \( M \) phones as:

\[
W_i = \{ p_{i1}(a_1), p_{i2}(a_2), \ldots, p_{iM}(a_M) \}
\]

where \( p_{im}(a_m) \) stands for the \( m \)-th phone segment with the corresponding phone identity \( a_m \), starting time \( t_m \) and ending time \( t_{m+1} \). Then, for every phone segment \( p_{im}(a_m) \) \( (1 \leq m \leq M) \), we compare it with the “true” phone segment generated from the force-alignment procedure. If \( p_{im}(a_m) \) doesn’t match well with the true segment, then we view it as a \( \text{rival} \) of the phone \( a_m \) and is assigned to the rival set \( \mathcal{S}(\neg a_m) \) accordingly. Otherwise, it is thought as the true observation of phone \( a_m \) and assigned to the set \( \mathcal{S}(a_m) \).

In this paper, the matching procedure between two phone segments is implemented as follows: if the hypothesized phone segment \( p_{im}(a_m) \) intersects its time registration with a phone segment with the same identity \( a_m \) in “true” phone segmentation, and the overlap between them exceeds 40% of total duration of the two segments, then we decide \( p_{im}(a_m) \) matches with the true label and \( p_{im}(a_m) \) is assigned to the set \( \mathcal{S}(a_m) \). On the other hand, if the hypothesized phone segment \( p_{im}(a_m) \) does not traverse with any phone segment with the same identity \( a_m \) in the transcription, and \( t_{m+1} > H_{m+1}^{\text{true}} \) and \( t_{m+1} > \xi \), where \( H_{m+1}^{\text{true}} \) denotes the average likelihood per frame of the hypothesis \( p_{im}(a_m) \) and \( H_{m+1}^{\text{true}} \) is the average likelihood per frame of the same segment based on the forced alignment procedure, then we consider the hypothesis \( p_{im}(a_m) \) as a \( \text{rival} \) of the phone \( a_m \) and it is assigned to \( \mathcal{S}(\neg a_m) \).

After we go through the whole database, we will have training data of the sets \( \mathcal{S}(a) \) and \( \mathcal{S}(\neg a) \) for every phone \( a \). In this work, positive and negative (rival) monophone HMM models are estimated for every phone based on maximum likelihood estimation (MLE). These monophone HMM models will be used only for utterance verification. They differ from the tri-phone HMM model used in our recognition experiments.

\[\text{A mechanism is implemented to guarantee that the same segment } p_{im}(a_m) \text{ will never occur in the same set more than once.}\]
4. A SINGLE-PASS STRATEGY FOR UTTERANCE VERIFICATION: VERIFICATION IN SEARCH

In most works on utterance verification, a 2-pass strategy is usually adopted. Recognition is performed first and the recognition results are passed to the 2nd-stage, utterance verification. In other words, utterance verification is performed on the recognized results as post-processing. In this case, recognized results usually is rejected if the scores are too low in the verification stage. Using this strategy, it is impossible to use utterance verification to correct some possible recognition errors made by the recognizer. Recently, some people have attempted to introduce some knowledge of utterance verification into recognition procedure in different ways, such as [1, 3]. In this paper, we propose to implement hypothesis-testing-based utterance verification during Viterbi search procedure. At every time instant \( t \), every partial word-ending path is treated as a null hypothesis, likelihood ratio testing is conducted for this path. If its score is below some threshold, this path will be rejected. In this way, a wrong path with high likelihood but low verification score probably can be rejected during search. We hope the correct path will thus emerge and recognition performance can be improved via this utterance verification procedure.

During the Viterbi search, at a time instant \( t \), suppose we have a partial active word-ending path \( \mathcal{L}(t) \), which is composed of a sequence of phone segments as \( \mathcal{L}(t) = \{a_1, a_2, \ldots, a_N\} \). Then a likelihood ratio score \( \eta \) is calculated for each phone segment based upon its positive and negative (rival) models as in Eq.(3). And the scores \( \eta \) of all phone segments in the entire path are averaged to get the final score for this path via the following standard method:

\[
CM(\mathcal{L}(t)) = \frac{1}{N} \sum_{i=1}^{N} \ln \eta(a_i) = \frac{1}{N} \sum_{i=1}^{N} \ln \frac{p(X_{i+1}^{i+1} | \Lambda_+ (a_i))}{p(X_{i+1}^{i+1} | \Lambda_- (a_i))}
\]

where \( X_{i}^{i+1} \) denotes the speech segment from time \( t_i \) to \( t_{i+1} \). If \( CM(\mathcal{L}(t)) < \tau \), where \( \tau \) is a pre-set threshold, the partial path \( \mathcal{L}(t) \) will be pruned out in the Viterbi search. Note that the above utterance verification is conducted for each active partial path independently at every time instant. Another advantage of the above method is that likelihood ratio based confidence measure is calculated and attached with every phone in all possible paths. These phone scores can be easily put together to get the confidence measures for word, phrase, or the whole sentence. And the final optimal path achieved in the Viterbi search also includes these confidence scores for its every phone components. These scores can be directly used for utterance verification in a post-processing stage as needed. Furthermore, the confidence measures can also be passed onto higher level modules, e.g., speech understanding.

The dynamic range of the likelihood ratio scores \( \eta \), computed in Eq.(3), is rather large. In most cases a confidence measure be improved via this utterance verification procedure. The intuitive explanation of \( \eta \) is the probability to be a true token of the phone \( a \) given the likelihood ratio value is \( \eta(a) \). Then the normalized score for a partial path is also similarly computed as:

\[
CM(\mathcal{L}(t)) = \frac{1}{N} \sum_{i=1}^{N} \eta(a_i).
\]

Obviously, \( CM(\mathcal{L}(t)) \) is a reasonable measure of the probability that the partial path \( \mathcal{L}(t) \) is correctly recognized.

Because likelihood ratio needs to be computed for all active paths at every time instant during the Viterbi search, one important implementation issue here is that we have to cache these verification scores efficiently to avoid recomputing them. We observed that the overall computation of likelihood ratio scores is kept under 5% of total computations when an effective caching is used.

5. EXPERIMENTS

To examine the viability of the proposed method, we evaluate on the DARPA Communicator task (travel reservation application) in the Bell Labs system. Utterance verification is incorporated into our Communicator system for the following purposes: i) to assist speech understanding and dialog management modules by providing some confidence measures for very recognized words; ii) to be able to detect and reject out-of-vocabulary words, e.g., unknown city names; iii) to improve the performance of speech recognition.

In our experiments, we are using three different data sets: i) Training set – it includes 6,232 utterances from travel reservation domain and a large amount of telephone speech data from other domains. It is mainly used to train acoustic models for recognition; ii) Development set – it is used for training positive and negative models for utterance verification, which includes 1,395 utterances; iii) Test set – used for evaluating the performance of both recognition and verification, including totally 4,001 utterances. These three different sets are collected for travel reservation application from different sites, and with a large number of different speakers.

In the baseline system, we used a 38-dimension feature vector, consisting of 12 Mel LPCcep, 12 delta CEP, 12 delta-delta CEP, delta log-energy and delta-delta energy. We use a state-tying, triphone HMM models as the recognition acoustic model. A class-based, tri-gram language model, including 2,600 words in total, is used in the system. This baseline system yields 19.4% word error rate (WER) in our test set.

To evaluate the performance of UV based on mono-phone rival model, we compare with the performance based on the normal mono-phone anti-model. Here the mono-phone anti-model represents the model trained from training and development data with fixed phone segmentation (generated from force-alignment), where all phone segments of the phone are collected to train the positive model for this phone and all other phone segments are used to train negative (anti-model) for this phone. Both positive and negative models trained in this way are called anti-models and they are used as the baseline utterance verification system. In this paper, both rival model and anti-models are trained from development set based on the maximum likelihood criterion.
5.1. To separate Phone tokens set $S_d$ from $S_r$.

In the first experiment, we investigate how well the estimated rival model can separate phone tokens assigned to $S_d$ and $S_r$. We first perform the training procedure presented in Section 3 on all utterances in the testing set to collect the sets $S_d$ and $S_r$ for all phones. Then we treat all tokens in $S_d$ as correct data and those in $S_r$ as imposters. The likelihood ratio test is conducted based on rival models and anti-models. The ROC curves are plotted in the Figure 1. From these curves, we see rival model derived scores, both CM1 and CM2, give better verification performance in all operating points and the normalized score CM2 is only slightly better than CM1 in some operating region.

5.2. To identify word errors in the recognizer’s output

The second experiments were to examine how well rival-model-based utterance verification can identify word recognition errors in the recognition results from our baseline system. We perform speech recognition for every utterances in our development and testing set. Then a confidence measure is calculated for every output word based on three different methods: i) Average of all its phone’s LLR (Log likelihood ratio) values computed based on the anti-model; ii) Average of all its phone’s LLR from rival model, denoted as CM1; iii) Normalized scores of CM1 as in Eq.(6), denoted as CM2. Based on these three different measures, we perform verification between correctly recognized words vs. misrecognized words (only substitution and insertion errors). The EER (equal error rate) for the development and testing sets are shown in Table 1. In development set, comparing with anti-model, CM1 improves EER from 37.9% to 22.3%. The big improvement indicates that our data selection procedure effectively collects rivals from development data set. As for test set, rival models still show some improvements in EER. CM1 and CM2 achieve 32.3% and 31.5% respectively, comparing with 40.0% got from anti-model. The same picture is also reflected in the ROC curves of testing set in Figure 2.

Figure 1: ROC curves of $S_d$ vs. $S_r$ for all phones in the testing set.

Figure 2: ROC curve of mis-recognized words vs. correctly recognized words in the recognition output of testing sets.

6. SUMMARY

In this paper, we have proposed a data selection method for utterance verification in continuous speech recognition. This data-driven method is used to train the so-called rival model for utterance verification based on likelihood ratio test. Some preliminary results on DARPA Communicator travel task have shown that rival models yield improvement of utterance verification performance over the classical anti-models.

ACKNOWLEDGEMENT

The authors thank Olivier Siohan and Jeff Kuo for their help to set up the baseline system.

7. REFERENCES