REJECTION USING RANK STATISTICS BASED ON HMM STATE SHORTLISTS

Enrico Bocchieri and S Parthasarathy

AT&T Labs-Research,
180 Park Ave., Florham Park, NJ 07932

ABSTRACT

We study a measure of confidence in the speech recognizer output based on a rank-order probability model of HMM state likelihoods. The motivation for rank models is based on the conjecture that statistics based on ranks are likely to be more robust than those based on the likelihood values, especially when the test and training distributions are mismatched. We investigate a number of different issues that arise in the development of rank models. We test the proposed rank-order model on two ASR rejection tasks: a combination of the log-likelihood ratio and rank order probability, yields relative reductions of the equal error rates of 31% and 8% (for the two tasks, respectively), over the log-likelihood ratio alone.

1. INTRODUCTION

Even though the accuracy of automatic speech recognition technology is not perfect under real-world conditions, speech recognizers are essential components in a growing number of interactive voice response and spoken natural language applications. An effective measure of confidence of an ASR hypothesis can improve overall user experience. Such confidence measures:

- enable the detection of erroneous recognizer hypotheses, or of out-of-grammar user input (*rejection* problem), and incorporation of these decisions in a dialog strategy.
- allow rejection of incorrectly recognized utterances, and thus prevent the contamination of the training data for unsupervised acoustic and language model adaptation schemes.

Confidence estimation is typically formulated as a hypothesis testing problem, where one must either accept or reject the hypothesis that the recognizer word output is correct. The confidence criterion is commonly based on one of the following:

(a) Acoustic model likelihoods, typically ratio of the likelihoods of the recognized hypothesis and of an alternate hypothesis. The alternate hypothesis may be represented either by a "backgound" or "garbage" model [1], or by a complex anti-model which is discriminatively trained [2, 3, 4].

(b) Word posterior probabilities derived from the recognizer outputs, such as wordgraph, lattices or sausages [5, 6, 7].

Our recognizer is based on context dependent phonetic (triphone) hidden Markov models (HMM), with tied states. A Gaussian mixture distribution is trained for each state in a state inventory. A context-dependent phone HMM is composed of HMM-states, each drawn from the state inventory. With the word *state*, we usually refer to an entry (tied state) in the state inventory, or, equivalently, to its corresponding Gaussian mixture.

Our approach to the problem of confidence measure is derived from the best path acoustic likelihoods as in (a). We investigate a metric based on the relationship between the likelihoods of a state and a set of close neighbors. This relationship is represented by a rank statistic described in the next Section. Similar rank statistics could be defined for larger units such as context-dependent or context-independent phones, or even a word. In a similar study, [8] has proposed the use of the state rank itself as a confidence measure, however, without an explicit rank probability model.

2. STATE CONFIDENCE MEASURES

Given a sequence of observation (frame) vectors $\mathbf{o}_1, ..\mathbf{o}_n$ (assumed independent), and the corresponding best HMM state sequence $s_1, ..s_n$ of the recognizer output (Viterbi state alignment), the confidence measure $C_{s_i}(\mathbf{o}_i)$ of frame $\mathbf{o}_i$ and state s_i may be defined as the logarithm of the likelihood ratio:

$$C_{s_i}(\mathbf{o}_i) \;=\; log\,(\,\frac{l_{s_i}(\mathbf{o}_i)}{l_{garb}(\mathbf{o}_i)}\,) \qquad (1)$$

where $l_{s_i}(.)$ and $l_{garb}(.)$ are the likelihood functions of state s_i and of the garbage model, respectively. Because of the independence assumption, adding (1) over the hypothesis frames gives the log-likelihood-ratio of the recognizer hypothesis.

The state likelihoods exhibit a large range of values even for correct recognition output: in fact the denominator in (1) partially compensates for such variability, and it is essential

for the performance of the state likelihood as a confidence measure. Intuitively, rather than the state likelihood value itself, a better confidence measure should use a characterization of the state hypothesis *relative* to the other competing HMM states. Such an approach has a discriminative flavor, and it should be more robust than those based on *absolute* measurements. This provides the motivation for this work. The general idea is as follows. For a given frame, $\mathbf{o}_i$, the likelihoods of all the states in the state inventory,

$$(\, l_{s_1}(\mathbf{o}_i), \, l_{s_2}(\mathbf{o}_i), \, ... \, l_{s_N}(\mathbf{o}_i))' \, , \tag{2}$$

are computed and sorted in decreasing order. The relative position (rank) of the j^{th} state among the N state likelihoods is denoted by r_{s_j}. We define a rank vector as follows:

$$(\, r_{s_1}(\mathbf{o}_i), \, r_{s_2}(\mathbf{o}_i), \, ... \, r_{s_N}(\mathbf{o}_i))' \tag{3}$$

The probability distributions of the rank vectors (3) given state s_i, correctly aligned with $\mathbf{o}_i$, are estimated from training data as in Section 3. During recognition, the test statistic for each frame $\mathbf{o}_i$ aligned with state s_i is defined as:

$$R_{s_i}(\mathbf{o}_i) \; = \; log \, (\, p_{s_i}(\, r_{s_1}(\mathbf{o}_i), \, r_{s_2}(\mathbf{o}_i), \, ... \, r_{s_N}(\mathbf{o}_i)) \tag{4}$$

where $p_{s_i}(.)$ is the probability of the rank vector (3) given hypothesis s_i.

3. RANK ORDER MODEL DESIGN

In principle, it is straightforward to implement non-parametric estimation of the rank models (4), by counting the occurrences of the rank vectors in the training data . A rank vector (3) is computed for every frame $\mathbf{o_i}$. By supervised Viterbi, we also find the state s_i aligned with $\mathbf{o_i}$, and we update the count of the rank vector of $\mathbf{o_i}$ for s_i. Then, count normalization yields an estimate of the rank probabilities.

In practice, this approach is not feasible, because of the large number of possible rank vectors ($N!$). Therefore, we assume independence of the state ranks in (4):

$$log \, (\, p_{s_i}(r_{s_1}(\mathbf{o}_i), ... \, r_{s_N}(\mathbf{o}_i)) \,) \; \approx \; \sum_{j=1}^{N} log \, (\, p_{s_i}(\, r_{s_j}(\mathbf{o}_i))) \tag{5}$$

The estimation of $p_{s_i}(r_{s_j}(.)) \; , \; i = 1, N$ in (5) still involves N^2 parameters, and hypothesis testing requires the computation and ordering of all N state likelihoods for every speech frame. Since in our models $N \approx 10^4$, (5) is still not practical.

We notice that in practice the rank distribution $p_{s_i}(r_{s_i})$ of the *correct* state is peaked around $rank = 1$ and that it tapers towards zero for ranks of about $16-64$. This suggests an additional simplification, which is to rank the likelihoods of *only* a shortlist of states that are the closest competitors of the correct state s_i. Intuitively, we remove from the rank order process those states that have little chance of confusion with the correct state. This is similar to the deletion of histogram bins with low counts, and to the assignment of their counts to the remaining bins. The *cohorts* for speaker verification follow a similar concept. In Section 4 we describe the shortlist selection algorithms. These produce, for every state s_i, a list of N' states ($s_j^i \; , \; j = 1, N' \; , \; N' << N$) sorted according to decreasing similarity to state s_i (obviously $s_1^i = s_i$). We can then estimate the probabilities $p_{s_i}(r_{s_j^i})$, where $r_{s_j^i}(.)$ denotes the rank of the likelihood of the j^{th} state in the shortlist ($s_j^i \; , \; j = 1, N'$) of neighbors of state s_i. Then instead of (5) we use:

$$R_{s_i}(\mathbf{o}_i) \; = \; \sum_{j=1}^{N'} log \, (p_{s_i}(\, r_{s_j^i}(\mathbf{o}_i))) \tag{6}$$

We also observed that the terms $p_{s_i}(\, r_{s_j^i}(.))$ in (6) are not useful in discriminating between correct and incorrect hypotheses if their distributions are "flat", and that the distributions for states s_j^i farther from the "correct" state s_1^i tend to be "flatter". Therefore, we limit the sum in (6) to the first M' terms:

$$R_{s_i}(\mathbf{o}_i) \; = \; \sum_{j=1}^{M'} log \, (p_{s_i}(\, r_{s_j^i}(\mathbf{o}_i))) \quad , \quad M' \; \leq \; N' \tag{7}$$

Experimentally, we found that $M' = 1$ gives good performance, and relatively small performance gains can be obtained for $M' \approx 4$.

4. SHORTLIST SELECTION

We have experimented with two techniques for the construction of a shortlist of neighbors for a given state s. The first technique is based on counting the number of times a state ranked in position 1 based on its likelihood, and the second technique is based on a similarity measure between states. The criterion based on first-rank counts is more coherent with the design described in Section 3, but it is computationally intensive in the training phase. However, it performed much better in preliminary experiments, and we have implemented it on parallel processors to achieve acceptable speed. In the experiments reported here, we always use state shortlists selected by first-rank counts.

4.1. First-rank counts

We keep a matrix of N by N counts $\mathbf{Q}$, where the s^{th} column is used to select the neighbors of the s^{th} state:

(a) For every frame $\mathbf{o}_i$ of the training data, find the state b_i with the highest (rank one) likelihood of all the HMM states, and the state s_i aligned with $\mathbf{o}_i$ (supervised Viterbi): then increment $\mathbf{Q}_{b_i,s_i}$ by one.

(b) For every state s, select its shortlist of N' states neighbors, as those with the largest values of the s^{th} column of $\mathbf{Q}$.

The criteria (ML or MMI) for HMM states estimation are not directly related to the state "rank" in step (a): it is not guaranteed that $\mathbf{Q}_{s,s}$ is the largest count in the s^{th} column of $\mathbf{Q}$, even though, as one may expect, this happens in the vast majority of columns (97%, in our case). Therefore, before (b), to ensure that the first state of the shortlist of neighbors of state s is s itself, we set to ∞ the diagonal elements of $\mathbf{Q}$.

4.2. State similarity metric

This method is similar to Section 4.1, except that $\mathbf{Q}_{s_j,s_i}$ is a measure of similarity between the j^{th} and i^{th} state:

$$\mathbf{Q}_{s_j,s_i} \;=\; l_{s_i}(\boldsymbol{\mu}_{s_j}) \;+\; l_{s_j}(\boldsymbol{\mu}_{s_i})$$

where $\boldsymbol{\mu}_{s_j}$ is the mean of the j_{th} state (from the state statistics), and $l_{s_i}(.)$ is the likelihood function of the i^{th} state.

5. MODEL TRAINING

We estimated the rank order model parameters on the same training data ($\approx 2\ 10^6$ words, telephone audio) of the recognizer HMM [9]. The frame vectors have 60 components, defined by a discriminative transformation (HDA) of 11 adjacent cepstral vectors. The HMM has 9,219 states with 92,100 Gaussians, discriminatively (MMI) trained. The rank probability models (7) are estimated, one for every HMM state, as in Sections 3 and 4.1.

6. REJECTION METHODS AND TASKS

After recognition of an input sentence, given the state path s_i of frames $\mathbf{o}_i$, we compute the rank confidence measure of the sentence hypothesis by:

- **Frame normalization**: arithmetic average (assuming independence) the frame confidence (7) over the hypothesis frames, or
- **State normalization**: compute the averages of the confidence (7) of consecutive frames aligned with the same HMM state, then average these state averages over the hypothesis. A measure similar to state normalization (phone normalization) is also in [10, 11].

To contrast the rank model (7) with the traditional likelihood ratio (1) , we have implemented the sentence hypothesis confidence measures above with the likelihood-ratio (1) as well.

We have also experimented with a sentence confidence measure defined by a simple linear interpolation of the rank-based and the likelihood-ratio metrics. The two interpolation weights are chosen to equalize the standard deviations of the rank-based and of the likelihood-ratio hypothesis confidence measures for in-domain data. This score combination gives better performance than either the rank-based metric or the likelihood-ratio metric alone.

We reject an utterance if its hypothesis confidence score is below a threshold, typically set to achieve the desired trade-off between false acceptances and false rejections. We determine the *equal error rate* by setting the rejection threshold to obtain the same rates of in-domain utterance rejections and out-of-domain acceptances, weighted by the the in-domain and out-of-domain sample counts, respectively. We report such an equal error rate for the following rejection tasks:

- **Account ID's**. The application is recognition of alphanumeric strings of various length (from 2 to 12), spoken after a prompt from an interactive system. The ASR word error rate for in domain-utterances is 3.5%. We want to reject the recognition hypothesis when the user does not pronounce an alphanumeric sequence. We test on 3,000 in-domain and 3,000 out-of-domain sentences.
- **Names**. The application is recognition of names (first and last) out of a 1,000 people directory, spoken after a prompt from a dialogue system. The string error rate for in-vocabulary names is 4%. We want to reject the recognition hypothesis when the user input is an out-of-vocabulary name. We test on 1,169 in-domain and 1879 out-of-domain utterances.

7. RESULTS AND DISCUSSION

Tables 1 and 2 show the equal error rates (eer) for the two rejection tasks, with the log-likelihood ratio and rank probability model as confidence measures. The rank model eer's are given for different values of the parameters N' and M' in (7). The equal error rate in parentheses are for the combination of the rank-based and likelihood ratio scores, as explained in the previous Section. For all the results shown, the sentence hypothesis confidence score is computed by state normalization (see previous Section). We have found that state normalization performs better than frame normalization for both the rank and the likelihood ratio scores. Intuitively, state normalizations gives the same weight to all phones, regardless of their average duration. Mismatched states are kept relatively short by the Viterbi alignment, so state normalization discriminates against the state alignment of incorrect hypotheses more than frame normalization.

The performance of combined rank probability and likelihood ratio scores is uniformly better than the performance using likelihood ratio scores alone, on both tasks and for a wide range of parameter values. For example, for $N' = 68, M' = 4$, the relative error reduction is 25% for the *Account ID's* task (from 5.05% to 3.78%), and 8% (from 17.3% to 16.0%) on the *Names* task.

The behavior of the rank probability method is somewhat different on the two tasks. On the *Account ID's* task, the confidence measure based on rank probability alone gives lower error rates (4.65%, 4.93%) than the likelihood ratio

(5.05%), for certain parameter values. On the *names* task, the error rate using rank models are about the same as that using likelihood-ratio scores for an appropriate choice of parameters. It is also interesting to note that the error rate continues to decrease as M' is increased. Other mechanisms for combining the rank probabilities of the candidate state and its neighbors may yield improvements.

These results based on rank-order probability models on the *Account ID* task are rather encouraging, because we obtain a significant reduction in the error rate, using the combined scores, on fairly difficult real application data. The overall rejection performance is probably good enough for real applications. Typical false acceptance (acceptance of out-of-grammar utterances as in-grammar utterances) cases are ones that are very short (1 or 2) alphanumeric sequences that are hypothesized in place of short out-of-vocabulary words (example: "e s" in place of "yes"). These can be detected based on a database lookup.

The *names* task appears to be much harder. Examples of out-of-vocabulary names that are recognized as an in-vocabulary name with a high score (and therefore falsely accepted) are: "Terri Slotterback" for "Mary Slotterback", "Mark Corgan" for "Mark Closson", "Ronda Newman" for "Ronda Bowman". It is clear that this is a difficult task because the differences are often very small. In many cases, a single phoneme substitution can map an out-of-vocabulary name into an in-vocabulary name. Another experiment was conducted with a larger grammar that included 137K names and the equal-error-rate increased to about 28%. This error rate is so high that it is not useful in practice. It is possible that out-of-vocabulary rejection would be more effective in these cases if we looked at local mismatches instead of a global utterance score. This is a subject of ongoing work.

Log likelihood ratio (%): 5.05

	Rank (%)	
	N'=64	N'=128
M'=1	5.48 (4.15)	4.93 (3.88)
M'=2	5.13 (3.91)	5.33 (4.06)
M'=4	4.65 (3.78)	5.38 (4.05)
M'=8	6.80 (4.38)	7.48 (4.81)

Table 1. Equal error rates for the *Account ID's* task. Results of combined rank and likelihood ratio in parentheses.

Log likelihood ratio (%): 17.3

	Rank (%)	
	N'=64	N'=128
M'=1	21.6 (16.5)	19.3 (16.3)
M'=2	21.3 (16.3)	18.1 (16.0)
M'=4	19.0 (16.0)	17.5 (15.9)
M'=8	18.1 (15.9)	16.6 (15.8)

Table 2. Equal error rates for the *Names* task. Results of combined rank and likelihood ratio in parentheses.

8. REFERENCES

[1] E.Lleida and R.C.Rose, "Efficient decoding and training procedures for utterance verification in continuous speech recognition," in *Proc. ICASSP'96*, 1996.

[2] R.A.Sukkar, A.R.Setlur, M.G.Rahim, and C.H.Lee, "Utterance verification of keyword strings using word-based minimum verification error (wb-mve) training," in *Proc. ICASSP'96*, 1996, pp. 516–519.

[3] R.A.Sukkar and C.H.Lee, "Vocabulary independent utterance verification for nonkeyword rejection in subword based speech recognition," *IEEE Trans. On Speech and Audio Proc.*, vol. 4, no. 6, pp. 420–429, November 1996.

[4] M.G.Rahim, C.H.Lee, and B.H.Juang, "Discriminative utterance verification for connected digit recognition," *IEEE Trans. On Speech and Audio Proc.*, vol. 5, no. 3, May 1997.

[5] F. Wessel, R. Schluter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Trans. On Speech and Audio Proc.*, vol. 9, no. 3, pp. 288–298, March 2001.

[6] 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*, pp. 14(4):373–400, 2000.

[7] D. Falavigna, R. Gretter, and G. Riccardi, "Acoustic and word lattice based algorithms for confidence scores," in *Proc. ICSLP'02*, 2002, pp. 1621–1624.

[8] Q. Lin, S. Das, D. Lubensky, and M. Picheny, "A new confidence measure based on rank-ordering subphone scores," in *Proc. ICSLP'98*, 1998.

[9] A. Ljolje, "Multiple task-domain acoustic models," in *Proc. ICASSP '03*, 2003, pp. 781–783.

[10] H. Bourlard and G. Bernardis, "Improving posterior based confidence measures in hybrid hmm/ann speech recognition system," in *Proc. ICSLP'98*, 1998, pp. 318–321.

[11] E. Mengusoglu and C. Ris, "Use of acoustic prior information for confidence measure in asr applications," in *Proc. Eurospeech'01*, 2001.