LVCSR LOG-LIKELIHOOD RATIO SCORING FOR KEYWORD SPOTTING

Mitchel Weintraub

SRI International
Speech Technology and Research Program
Menlo Park, CA, 94025

ABSTRACT

A new scoring algorithm has been developed for generating wordspotting hypotheses and their associated scores. This technique uses a large-vocabulary continuous speech recognition (LVCSR) system to generate the N-best answers along with their Viterbi alignments. The score for a putative hit is computed by summing the likelihoods for all hypotheses that contain the keyword normalized by dividing by the sum of all hypothesis likelihoods in the N-best list. Using a test set of conversational speech from Switchboard Credit Card conversations, we achieved an 81% figure of merit (FOM). Our word recognition error rate on this same test set is 54.7%.

1. INTRODUCTION

This paper describes how SRI International has applied DECIPHER™ to the keyword-spotting task. DECIPHER™ is SRI's large-vocabulary speaker-independent continuous-speech recognition (CSR) system, which we have used for a number of different CSR tasks, including Resource Management (RM), Air Travel Information Systems (ATIS), and *Wall Street Journal* (WSJ) dictation [1-4].

A number of other hidden Markov model (HMM) based systems have previously been developed for keyword spotting [5-8] which has two important dimensions:

- How keywords are hypothesized
- How keyword scores are computed

In our earlier work on keyword spotting [8], we used the Viterbi backtrace from a large-vocabulary continuous speech recognition (LVCSR) system. A keyword was hypothesized if it appeared in the Viterbi backtrace. Using the one best answer from the Viterbi backtrace, we used the average probability per frame as the score for each hypothesized keyword.

This algorithm worked well for high frequency keywords, but was not able to generate the necessary false-alarms (when the recognizer inserts this keyword) needed to compute an receiver-operating curve (ROC).

To improve keyword spotting performance, we need to increase the frequency that keywords are hypothesized. To complement this goal, we need a scoring algorithm that will continue to reward hypotheses that are the best recognition hypothesis.

The score used for hypothesizing keywords in our earlier this system was based on a duration normalized likelihood:

$$\frac{\text{Score}_{Keyword}\Big|_{StartTime_{Keyword}}^{EndTime_{Keyword}}}{EndTime_{Keyword} - StartTime_{Keyword}} \quad (1)$$

In contrast to the duration normalized likelihood, many other researchers have developed and use a log-likelihood ratio metric [5,6]:

$$\text{Score}_{Keyword} - \text{Score}_{BackgroundModel}\Big|_{StartTime_{Keyword}}^{EndTime_{Keyword}} \quad (2)$$

The advantage of such an approach is that the quality of the match to the data is not directly used, but the comparison is relative to how it matches other candidate hypotheses. However this likelihood ratio only uses acoustic information about the keyword hypothesis.

One of the central aspects of our method is an extension of this approach (Eq. 2) to an LVCSR system. In order to incorporate additional information (primarily language modeling information about the sequence of words containing the keyword), we incorporate this technique into a LVCSR system. The proposed metric (described in Section 2) is called LVCSR log-likelihood ratio scoring to denote the similarity to the above idea.

Another approach to computing a keyword hypothesis score has been developed in [7]:

$$Prob\,(\,\text{State}_{Time\,=\,t} = EndState_{Keyword}\ |\ \text{Observations}) \quad (3)$$

While this approach also has the potential for using a large-vocabulary CSR approach, the scoring metric has several disadvantages. By looking at the last state in the

0-7803-2431-5/95 $4.00

keyword, we distinguish this state from all the other states in the word hypothesis. If the last state does not match the data well (even though all the other states have matched well), then this focus on how a word ends can degrade system performance.

2. LVCSR LOG-LIKELIHOOD RATIO SCORING

2.1. Approach

The new scoring metric we propose is:

$$\frac{\sum_{\vec{W}:\,(KW \in w_i)} P\left(\{w_1, \ldots, w_n\} \mid \text{Obs}\right)}{\sum_{\vec{W}} P\left(\{w_1, \ldots, w_n\} \mid \text{Obs}\right)} \quad (4)$$

where:

$\vec{W}$ is the word sequence: $\{w_1, \ldots, w_n\}$

$\vec{W}: (KW \in w_i)$ is all word sequences that contain the keyword KW.

For a given set of observations (Obs) and a set of HMM recognition models, we can compute a probability distribution over all word sequences. The numerator in Eq. 4 is the sum of all such word sequences that contain the keyword, while the denominator is the sum of the probability of all word sequences.

The ratio of these two quantities is the percentage of all recognition hypotheses (weighted by the probability of different sequences) in which the keyword appears. If a keyword appears in all likely word sequences, then it will have a LVCSR log-likelihood ratio score of 0.0 (equivalent to the log of the probability ratio of 1.0).

Using Eq. 4, a keyword would be hypothesized whenever there is a non-zero probability of a word sequence that contains the keyword. The remaining issues to resolve are:

- How to associate times (start, end) with keywords
- How to hypothesize and score a single keyword that appears multiple times in a single sentence
- How to implement Eq. 4 using an LVCSR system

2.2. Implementation

Our implementation of Eq. 4 uses N-best lists [9, 10]. The N-Best lists correspond to the word sequences $\vec{W}$ that we will be searching for the keyword.

We compute the N-best lists using a progressive search approach [2]. First, an initial two-pass recognition system is used to generate word-lattices. Then, another two-pass recognition system is used to generate N-best lists using the word lattices to prunes the search.

If a keyword appears anywhere in the N-best list, it will be hypothesized, with a score computed based on Eq. 4 as follows:

$$\frac{\sum_{\vec{W}:\,(KW \in NB_i)} P\left(\vec{W}\right) \cdot P\left(Obs \mid \vec{W}\right)}{\sum_{\vec{W}} P\left(\vec{W}\right) \cdot P\left(Obs \mid \vec{W}\right)} \quad (5)$$

where:

$P\left(Obs \mid \vec{W}\right)$ is the acoustic HMM probability

$P\left(\vec{W}\right)$ is the language model probability

$\vec{W}: (KW \in NB_i)$ is the list of all N-Best word sequences that contain the keyword

In comparing several recognition hypotheses, it is important to resolve hypotheses where a word can appear multiple times. To allow for this condition, each N-best hypothesis contains the timing information of the Viterbi backtrace associated with each word. This information is used as follows:

- If the same keyword appears multiple times in a single recognition hypothesis, it is considered as separate keyword hypotheses.
- If two keyword hypotheses (each from a different N-best recognition hypothesis) overlap in time, they are considered to be the same keyword hypothesis.
- As we proceed through the N-best list (from best to worst hypotheses), the time alignment for a particular keyword hypothesis (start, end) uses the time alignments from the hypothesis with the highest (probability per frame) feature.

Finally, we modified the implementation of Eq. 5 for breaking ties. If a keyword appears in all the N-best lists, then it will receive a score of 0.0 (log(1.0)). To run NIST's credit-card wordspotting software, it is important to break ties between false-alarms and true-hits (otherwise the false alarm is assumed to be of a higher score). Therefore, we have modified the implementation of Eq. 5 to add in an epsilon weight times the score in Eq. 1.

3. EXPERIMENTAL RESULTS

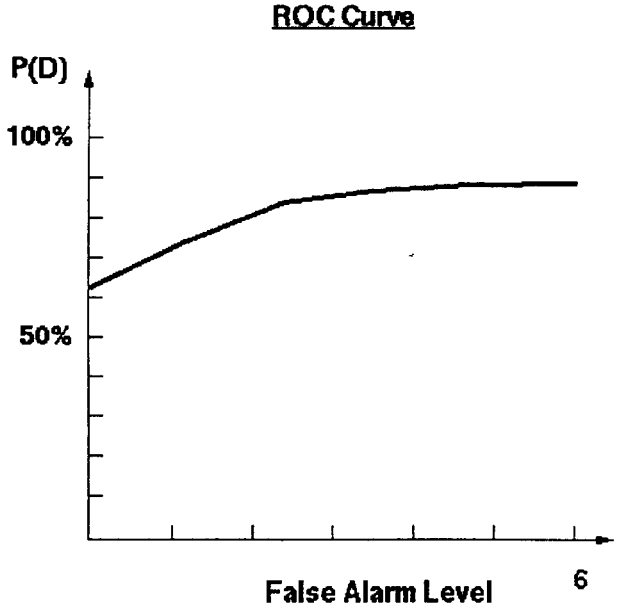
3.1. Development Test Set Description

A development test set has been assembled for recognition and wordspotting experiments. It contains the same speakers and conversations that were used at the 1993 Robust Recognition Workshop in Rutgers, New Jersey. All the conversations were subdivided using their Switchboard [11] .mrk files, and an expert transcriber corrected all the transcriptions by hand. This test set contains 10 male and 7 female speakers, for a total of 1,928 utterances (1,065 male, 863 female). Hand-corrected keyword references (.ref files) were generated so that this same testset can be used for developing wordspotting technology.

3.2. Experiments

The speech recognition system used a vocabulary size of 5,000 word which included all the keywords. A bigram grammar was used as well as genonic HMM acoustic models [1]. The size of the N-best lists used were 500. The two-pass system that generates lattices used a lexical-tree for the back-off node [12].

The wordspotting ROC curve for using the LVCSR log-likelihood is shown below. The probability of detection (P(D)) is plotted as a function of the number of false alarms. This ROC curve corresponds to an average probability of detection (averaged from 0 to 10 FA/KW/HR) of 81% (FOM).



To compare this algorithm to other approaches, we tested the performance of several algorithms. Those results, displayed in Table 1, are also on the credit card subset of the Switchboard corpus.

Using our previously developed Viterbi 1-best path algorithm [8], we achieved an FOM of 69.7%. The limiting factor in this approach is that it does not hypothesize enough false alarms.

The Viterbi 1-best approach can be extended to hypothesize a keyword any time the keyword appears in an N-best list. The score of this keyword is the best probability-per-frame score of any hypothesis in any of the N-best lists. As the very poor performance (41.5% FOM) of this algorithm shows, many false alarms that appear in other locations in the N-best list that will have a good score (probability per frame), even though the overall recognition hypothesis that this keyword appears in received a very poor score.

The last entry in Table 1 is the new LVCSR log-likelihood ratio metric described in Section 2. The LVCSR log-likelihood approach shows an improved detection rate at high false alarm rates, while maintaining an even higher average probability of detection (FOM of 81%).

Wordspotting Algorithm	Figure of Merit
Viterbi 1-Best Hypothesis Using Duration-Normalized Likelihood	69.7
N-Best Hypothesis Using Duration-Normalized Likelihood	41.5
LVCSR Log-Likelihood Ratio Scoring	81.0

Table 1: Credit-Card FOM for different wordspotting algorithms

4. DISCUSSION

A new scoring algorithm has been implemented for spotting keywords using a large-vocabulary continuous speech recognition system. This technique uses the N-best answers and their Viterbi alignments to compute the probability that each particular keyword is present in an utterance. The score for a putative hit is computed by summing the likelihoods for all hypotheses that contain the keyword and dividing by the sum of all the likelihoods for all the hypotheses in the N-best list. In cases where the keyword exists in all the N-best answers, this score will be 0.0 (log probability of 1.0).

We have contrasted this keyword scoring algorithm to earlier approaches, and presented experimental evidence of how this algorithm is superior to earlier approaches.

One of the factors that has led to this improvement is the incorporation of additional knowledge (language

modeling) in an LVCSR framework. The results that we have presented used a bigram language model trained on 2 million words of conversational speech. If we had switched domains to a different type of input where the language model was no-longer a good match to the observed data, it is not clear how much additional improvement this technique will add. Rapid adaptation of language model is a current area of research that is aimed at solving this limitation.

The N-best implementation is straightforward and has the advantages that additional knowledge sources can be easily incorporated into the scoring algorithm (e.g. word and phone duration modeling, N-gram language models). However, we have found that for conversations speech (with a high word-error rate), that there are significant number of times when the correct word does not appear in the N-best list. For applications that require high probability of detection with corresponding high false-alarm rates, then a direct search of a word-lattice [2] might lead to a better implementation of the above algorithm.

REFERENCES

1. V. Digalakis, and H. Murveit, "Genones: Optimizing the Degree of Mixture Tying in a Large Vocabulary Hidden Markov Model Based Speech Recognizer," 1994 *IEEE ICASSP*, pp. I537-I540.
2. H. Murveit, J. Butzberger, V. Digalakis, and M. Weintraub, "Large-Vocabulary Dictation Using SRI's DECIPHER Speech Recognition System: Progressive Search Techniques," 1993 *IEEE ICASSP*, pp. II319-II322.
3. H. Murveit, J. Butzberger, and M. Weintraub, "Performance of SRI's DECIPHER Speech Recognition System on DARPA's CSR Task," 1992 DARPA Speech and Natural Language Workshop Proceedings, pp 410-414
4. H. Murveit, J. Butzberger, and M. Weintraub, "Reduced Channel Dependence for Speech Recognition," 1992 DARPA Speech and Natural Language Workshop Proceedings, pp. 280-284.
5. B.P. Landell, R.E. Wohlford, and L.G. Bahler, "Improved Speech Recognition in Noise," 1986 *IEEE ICASSP*, pp. 749-751.
6. R. Rose and D. Paul, "A Hidden Markov Model Based Keyword Recognition System," 1990 *IEEE ICASSP*, pp. 129-132.
7. R. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous Hidden Markov Modeling for Speaker-Independent Word Spotting," 1989 *IEEE ICASSP*, pp. 627-630.
8. M. Weintraub, "Keyword-Spotting Using SRI's DECIPHER Large-Vocabulary Speech-Recognition System," 1993 *IEEE ICASSP*, pp. II463-II466.
9. R. Schwartz, "Efficient, High-Performance Algorithms for N-Best Search," 1990 DARPA Speech and Natural Language Workshop, pp. 6-11.
10. F.K. Soong, E.F. Huang, "A Tree-Trellis Based Fast Search for Finding the N Best Sentence Hypotheses in Continuous Speech Recognition," 1990 DARPA Speech and Natural Language Workshop, pp. 12-19.
11. J.J. Godfrey, E.C. Holliman, and J.McDaniel, "SWITCHBOARD: Telephone Speech Corpus for Research and Development," 1992 *IEEE ICASSP*, pp. I-517-I-520.
12. H. Murveit, P. Monaco, V. Digalakis, and J. Butzberger, "Techniques to achieve an accurate real-time large-vocabulary speech recognition system," 1994 ARPA Human Language Technology Workshop, pp. 368-373.