RECOGNITION-BASED WORD COUNTING FOR RELIABLE BARGE-IN AND EARLY ENDPOINT DETECTION IN CONTINUOUS SPEECH RECOGNITION

Anand R. Setlur and Rafid A. Sukkar

Lucent Technologies
2000 N.Naperville Rd., Naperville, IL 60566, USA
{asetlur, sukkar}@lucent.com

ABSTRACT

In this paper, we present a word counting method that enables speech recognition systems to perform reliable barge-in detection and also make a fast and accurate determination of end of speech. This is achieved by examining partial recognition hypotheses and imposing certain "word stability" criteria. Typically, a voice activity detector is used for both barge-in detection and end of speech determination. We propose augmenting the voice activity detector with this more reliable recognition-based method. Experimental results for a connected digit task show that this approach is more robust for supporting barge-in since it is less prone to interrupting the announcement when extraneous speech input is encountered. Also, by using the early endpoint decision criterion, average response times are sped up 75% for this connected digit task.

1. INTRODUCTION

Voice activity detectors (VADs) are used in continuous speech recognition applications to determine speech events. They are used to mark the beginning and end of a spoken set of words. These detectors, however sophisticated, have the limitation of not being able to differentiate between in-vocabulary and out-of-vocabulary speech since their decisions are based solely on energy magnitude and duration. In typical speech recognition applications, there is a system prompt that solicits speech input from the user. New users typically listen to the entire system prompt prior to responding, whereas experienced users interrupt the prompt by speaking over the prompt (referred to as barge-in). Most speech applications support barge-in by using a VAD to detect the onset of speech input and disabling the prompt when speech is detected. However, supporting a reliable barge-in scheme is a challenging issue. Recently, some research efforts have been directed towards making barge-in more robust[1][2]. Ideally, we would want to interrupt the prompt only if a user starts to speak valid, in-vocabulary speech and not interrupt the prompt for invalid speech inputs which may be coughs, breath sounds or out-of-vocabulary words. A voice activity detector cannot be used for reliable barge-in since it will interrupt the announcement without determining if the initial speech segment corresponds to in-vocabulary speech. In this paper, we will present a method that examines partial recognition hypotheses from a speech recognizer to make a decision that a valid keyword exists in the speech utterance. While the VAD-based barge-in detector will trigger on most every extraneous speech or noise event, a recognition-based barge-in detector, which we will discuss in this paper, is significantly more reliable in these cases.

The other aspect that this paper addresses is improvement of average response times in known-wordlength continuous speech recognition. Response time is the time elapsed from the end of speech input to when the recognition result is obtained by the recognizer. By using only a VAD to detect the end of speech, the response time is at least greater than the inter-word gap time of the VAD, even if the recognition algorithm runs in real-time. The reason for this is that the inter-word gap time needs to elapse with no speech activity before the VAD can declare end of speech input. In this paper, we propose a scheme to detect the end of a speech utterance sooner than the time it takes a VAD to determine the endpoint. This is achieved by examining partial recognition hypotheses and counting the number of words in each path. If all the viable paths have "stabilized" to a point where no new hypotheses are likely to be introduced, an endpoint decision is made. Using recognition to make this endpointing decision is reliable and, on average, the resulting endpoint occurs significantly sooner when compared to the VAD endpoint. The additional benefit from a faster response time is that the recognition resource is freed up earlier to process the next request. This means that we can, on average, process more calls or make do with less computational resources.

The organization of the paper will be as follows: In the next section, we will give a brief overview of the recognizer that we use in our system and discuss how the recognition hypotheses are stored in a decoding tree. In section 3, we describe the word counting algorithm that periodically examines the decoding tree and discuss its use for barge-in and early endpointing. Experimental results relating to response times for barge-in and endpointing on a connected digit task are given in section 4 followed by conclusions in section 5.

2. OVERVIEW OF THE RECOGNITION SYSTEM

The recognition system that we use is a frame-synchronous beam search algorithm [3] that employs the wave decoder described in [4]. Twelve LPC-derived cepstral coefficients, normalized energy and their first and second order derivatives constitute the 39-element feature vector [5]. The feature vector is updated every 10 ms and is computed over a 30 ms window. Acoustic speech events are modeled as continuous density hidden Markov models. Most of the models are dedicated to modeling in-vocabulary keywords. However, a handful of "filler" models attempt to model out-of-vocabulary speech events. In order to support wordspotting, the grammar allows filler (also known as garbage) words to optionally precede and follow keyword speech.

The recognition problem boils down to a search for the most likely (highest likelihood score) word sequence $w_1, w_2, ..., w_n$

that best explains the input speech feature vector sequence under certain grammar constraints. A word network detailing which words can precede and follow which words is compiled from the grammar specifications. A phone network that details which phones can precede and follow which phones is then derived from the word network. The algorithm uses Viterbi decoding to find the optimal phone sequence under the specified grammar constraints. A full search of all possible phones in the network to find the best phone sequence is too large and a beam search significantly reduces the search space and lends itself well to practical implementations. In a beam search, only those phone sequences that are likely (i.e., have likelihood scores within a prescribed difference from the current best score) are retained and extended. Unlikely hypotheses are pruned from the search space.

At the start of a speech utterance, only valid start phones as specified in the phone network are marked as active. At each time frame, t, dynamic programming using the Viterbi algorithm is performed only over the active portion of the phone network. The active portion of the phone network varies with time since we employ a beam search strategy. All newly extended phones get added to the active portion of the network and pruned phones get deleted. The wave decoder [4] aims to restrict dynamic memory usage to a minimum by allocating space for only the active portion of the network instead of the entire network. It also reclaims space from the portion of the network that becomes inactive.

To be able to retrieve the phone sequence that corresponds to the winning cumulative likelihood score, we need to store the partial phone sequences in a linked list fashion; one linked list per viable phone sequence. This set of linked lists constitutes the decoding tree. Each entry in the decoding tree (termed DTENTRY) is associated with a specific phone in some viable phone sequence. Each DTENTRY contains information regarding the frame number when the phone was first activated and pointers to the preceding and following DTENTRYs of the sequence. The decoding tree is updated every frame to reflect any changes in the set of viable phone sequences that lie within the beam. A link is maintained between the cumulative likelihood score for each active phone which is part of some viable sequence of phones and the most recent DTENTRY associated with that sequence. Using this information, one can backtrack through the phone sequence that any surviving path took from the start to the current time instant.

Typically, a VAD is used to determine when to start and stop processing speech input. Once the end of speech marker is set by the VAD, backtracking is performed to pick the winning string and involves traversing the path in the decoding tree with the highest cumulative score. In this paper, we propose to examine the contents of the decoding tree periodically, instead of only once for backtracking at the end, to determine if a valid barge-in has occured and also to perform early endpoint detection.

3. WORD COUNTING PROCEDURE

At the start of a word, the decoding tree is very fuzzy in the sense that there are several viable phone sequence hypotheses. Gradually, as we progress deeper into the word, fewer of the hypotheses survive due to the beam search strategy that we employ, until the point where there are only a handful of viable hypotheses that explain the spoken word. However, when the speech input is out-of-vocabulary, the decoding tree continues to remain fuzzy since none of the word models will match the input well. In most instances, if the word is modeled well and the input speech matches well with the model, we are left with only one DTENTRY that represents that segment of time and it is part of every viable path. This is illustrated in Figure 1 where at the start of the word, there are many hyotheses active but by the beginning of the next word there are very few DTENTRYs active for the segment of time corresponding to the previous word.

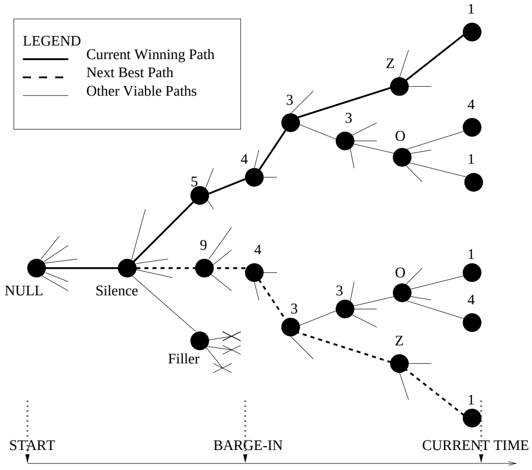

Figure 1: A Sample decoding tree evolving over time

We conclude from the above discussion that uncertainty present at the start of a spoken word dissipates by the time we reach the later words in the sequence. By periodically examining the contents of the decoding tree, and tracing back through all viable paths at the current time instant, one can determine how many words have stabilized across all paths. Traversing through the decoding tree every frame does add roughly a 4% overhead in our connected-digit recognition implementation and is an overkill. So we chose to examine the decoding tree every 10^{th} frame to look for newly stabilized words and reduced this overhead to 0.4% in the process. Two applications of word counting that are of particular interest are recognition-based barge-in and early endpointing. In the case of barge-in, we want to interrupt the announcement as soon as we know that the first word has been spoken. In contrast, for the early endpointing decision, we would like to stop processing as soon as the last spoken word has stabilized.

Figure 2 shows the algorithm in action on a sample 14-digit utterance. First and last digit end times are marked by solid lines. The two dashed lines next to the solid lines respectively represent the time instants when barge-in and early end decisions were reported. For this example, barge-in is reported about 180 ms after the first digit ended and an early endpoint is reported about 480 ms after the last digit ended. Notice that the barge-in did not falsely trigger on the initial noise segment that occured before the first digit was spoken but instead waited until after the first digit ended. This demonstrates that the recognition-based barge-in is more robust compared to a VAD-based barge-in scheme that cannot distinguish between digits and other speech events.

3.1. Recognition-Based Barge-In

Recognition-based barge-in is the idea of using a recognizer to determine when the first in-vocabulary word was spoken and subsequently cutting off any announcement that the system may have

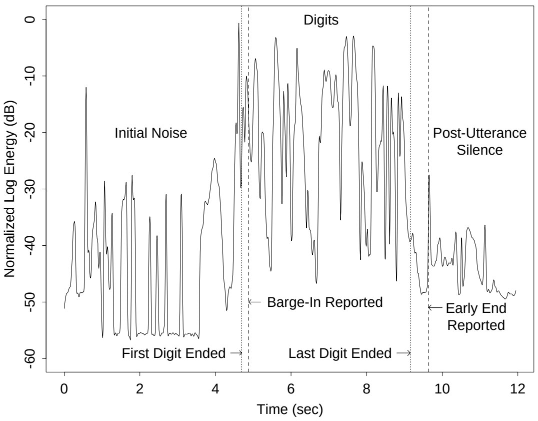

Figure 2: A sample 14-digit utterance

been playing at that point in time. When user input is a cough or breath or any other out-of-vocabulary speech event, there is a good chance that the announcement will not be interrupted since the speech segment is likely to match better with a filler model compared to other models. This ability to continue playing the announcement when extraneous speech is encountered is a desirable attribute and is the main advantage of this approach compared to its VAD-based counterpart.

Filler and silence segments are considered contentless whereas in-vocabulary words have word content associated with them. For determining the barge-in decision point, we periodically examine the decoding tree and insist that every viable path in the decoding tree have a word with content associated with it. This also means that at the barge-in decision point, there is is no longer any viable path consisting solely of contentless words. So the moment that all the paths in the decoding tree have at least a single word of content (i.e, not solely filler or silence) associated with them, a barge-in decision point is reached. Figure 1 shows that the barge-in decision point is typically declared as soon as the contentless paths becomes inactive which could, in some instances, be into the middle or end of the second word.

3.2. Recognition-Based Early Endpoint Detection

For known-wordlength recognition applications, where the number of expected words is pre-determined, it would be desirable to terminate the recognition process as soon as the expected number of words have been detected. This attribute is desirable both in terms of fast recognition response times and also in terms of minimizing resource usage. Typically, connected digit applications such as account number or telephone number recognition are known-wordlength tasks.

Similar to the recognition-based barge-in detection case, we periodically examine the decoding tree contents and skip over the segments that have no word content associated with them. This time, however, we insist that the final word "stabilize" on all the paths. Stability is satisfied by checking for the ending times of the last word in each of the viable paths and insisting that the last word with content end at the same frame number in each of the paths. Synchronization of ending frame number across all viable paths is a stiff requirement. Nevertheless, it is satisfied often enough and early into the post-utterance silence portion. There are, however, a fraction of speech utterances for which this stringent requirement is not satisfied. For such cases, a VAD endpoint is used to terminate the recognition process. Therefore we propose using the recognition-based early decision algorithm in parallel with a VAD.

The endpoint decision marker can be set by either the early endpointing method that we have outlined above or after the gap timer has expired in the VAD, whichever is first. This ensures that the worst-case response time of the parallel system is no worse than that obtained by the VAD-only endpointing scheme. In general, as will be evident in the next section, the average response times are much improved.

4. EXPERIMENTAL RESULTS

The task that was chosen to measure the effectiveness of the algorithms outlined in this paper is a connected digit application. The testing database consists of 15000 connected digit strings collected over a variety of telephone connections. The evaluation was done using known-wordlength grammars with a different grammar being chosen depending on the length of the string. Digit lengths varied from 1 to 16 with the average digit length being 6. Two filler models, one modeling all types of non-keyword speech and sounds and one modeling breath were used in conjunction with silence and 275 context-dependent head-body-tail models for the digits 1 through 9, Z (zero) and O (oh). More information on this model topology can be found in [6].

Figure 3 shows a histogram of the time difference between when the barge-in decision was made and when the first word of the winning path actually ended. The word ending time of the first word for this purpose is determined at the end of the utterance by tracing back through the path that had the best cumulative score. As is evident from Figure 3, the average delay in reporting barge-in is about 130 ms from the end of the first digit and typically occurs into the second digit of a connected digit utterance. There are some cases where a decision is made even before the first word has ended and are represented in the lower left corner of the histogram. There are roughly 1.5% of utterances not represented in this histogram for which barge-in decision could not be made since some contentless path remained viable for the duration of the utterance. This is a trade-off that has to be made if one desires to selectively barge-in based on the recognition decision.

To evaluate out-of vocabulary performance, a database of 6600 utterances consisting of short non-digit phrases spoken by a variety of speakers was selected and an unknown length grammar was used. When the recognizer was presented with short non-digit (out of vocabulary) phrases, it did not report barge-in on roughly 82% of the sentences and wrongly triggered on the remaining 18%. This is still substantially better than a VAD-based scheme which would have wrongly triggered on 100% of the non-digit phrases. One can augment the recognition-based barge-in decision module that relies on filler models to filter out non-digit utterances, with an utterance verification module [7], which will further lower the rate of incorrect barge-in when out-of-vocabulary speech is encountered.

Figure 4 shows a histogram of the time difference between when the early-endpointing decision was made and when the last word actually ended. The actual ending time of the last word is again determined by the backtracking that is done at the end on the winning path. We can see that the average delay in reporting

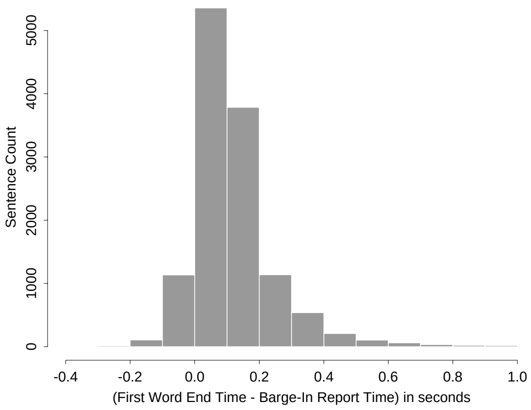

Figure 3: Histogram of time elapsed before barge-in is reported

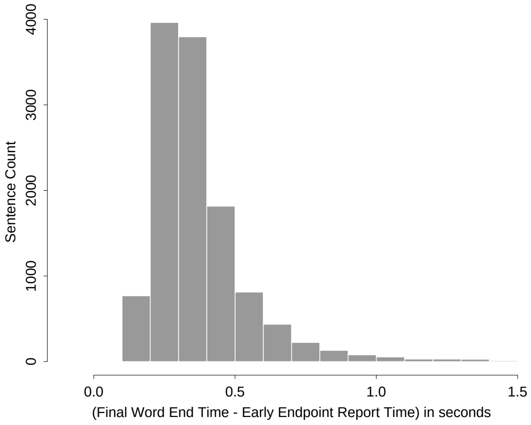

Figure 4: Histogram of time elapsed before endpoint is reported

the end in relation to when the last word actually ended is about 375 ms. This represents a 75% improvement in average response time when compared to a gap-time of 1500 ms that has to expire before any decision can be made in the VAD-only case. For connected digit tasks, typical gap-time values are between 1000-1500 ms. One caveat is that for about 2.5% of the utterances an early endpointing decision could not be made since the word end did not stabilize prior to the gap-timer expiring. For these utterances we have to rely on the VAD to detect the end of the utterance so backtracking can be performed to determine the winning sentence. In the rest of the cases, backtracking was performed as soon as the early endpointing decision was made. Since the early endpointing decision is based on a sufficiently stringent criterion, no new errors are introduced by abandoning the search early. Therefore, the recognition accuracy remains the same whether we used a VAD-only scheme or a VAD in parallel with the recognition-based early end decision scheme.

Only 8% of the correctly recognized sentences had response times of greater than 0.75 seconds whereas 34% of the incorrectly recognized sentences had response times of over 0.75 seconds. So a higher proportion of errors have longer response times compared to correctly recognized strings. This fact can possibly be exploited to aid in distinguishing between correct recognitions and misrecognitions and is a topic for future work..

We notice from Figure 4 that the decision to terminate is always after the last digit has ended, unlike in Figure 3 where a barge-in decision was made in some cases prior to the first word ending. This shows that the early endpointing decision criterion that we have chosen is more conservative when compared to our barge-in criterion. Making the termination criterion weaker may result in quicker response times but will also in general lower accuracy by terminating too soon, earlier than the end of the last word in some instances.

5. CONCLUSIONS

We have presented a method to perform barge-in and early endpoint detection using information already available in the recognizer. Recognition-based barge-in is more robust to extraneous sounds compared to VAD-based barge-in. The recognition-based early end decision scheme that we also presented has practical significance in that it speeds up recognition response times and uses fewer computational resources in the process. Experimental results on connected digits demonstrate the effectiveness of these schemes compared to using only a VAD for purposes of barge-in and endpoint detection. The barge-in criterion can be further strengthened by adding an utterance verification component to this system and is an ongoing topic of interest in our research.

6. REFERENCES

[1] E. Bauche, B. Gajic, Y. Minami, T. Matsuoka and S. Furui, “Connected digit recognition in spontaneous speech,” *Proc. Eurospeech ’97*, Vol. 2, pp 923-926, Sept 1997.

[2] S. Yamamoto, M. Naito and S. Kuroiwa, “Robust speech detection method for speech recognition system for telecommunication networks and its field trial,” *Proc. Eurospeech ’97*, Vol. 3, pp 1535-1538, Sept 1997.

[3] C. H. Lee and L. R. Rabiner, “A frame-synchronous network search algorithm for connected word recognition,” *IEEE Trans. on Acoustics Speech and Signal Processing*, Vol. 37, No. 11, pp 507-520, Nov 1989.

[4] W. Chou, E. Buhrke and Q. Zhou, “A wave-decoder for continuous speech recognition,” *Proc. ICSLP ’96*, pp 2135-2138, Oct 1996.

[5] J. G. Wilpon, C. H. Lee and L. R. Rabiner, “Improvements in connected digit recognition using higher order spectral and energy features,” *Proc. ICASSP ’91*, Vol 1, pp 349-352, May 1991.

[6] W. Chou, C.-H. Lee, B.-H. Juang, “Minimum error rate training of inter-word context dependent acoustic model units in speech recognition,” *Proc. ICSLP ’94*, pp 439-442, Sept 1994.

[7] 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,” *Proc. ICASSP ’96*, Vol 1, pp 518-521, May 1996.