A Rapid Match Algorithm for Continuous Speech Recognition

Laurence S. Gillick and Robert Roth
Dragon Systems, Inc.
90 Bridge St.
Newton MA. 02158

Abstract

This paper describes an algorithm for performing rapid match on continuous speech that makes it possible to recognize sentences from an 842 word vocabulary on a desktop 33 megahertz 80486 computer in near real time. This algorithm relies on a combination of smoothing and linear segmentation together with the notion of word start groups. It appears that the total computation required grows more slowly than linearly with the vocabulary size, so that larger vocabularies appear feasible, with only moderately enhanced hardware. The rapid match algorithm described here is closely related to the one that is used in DragonDictate, Dragon's commercial 30,000 word discrete utterance recognizer.

1. Introduction

For it to be feasible to perform large vocabulary continuous speech recognition in near real time on currently available, moderately priced hardware, computational compromises appear to be essential. One obvious compromise is to avoid detailed consideration of certain hypotheses that "cursory" inspection would reveal to be exceedingly unlikely. But what constitutes cursory inspection? To put it somewhat differently, how can we perform a very quick computation that would allow us to throw out many of the hypotheses that are "obviously" false? In particular, need dynamic programming play a role in such a "rapid match" algorithm?

In this paper we shall describe a strategy for performing this kind of computation in the context of continuous speech recognition. The algorithm is an extension of the rapid match procedure that is used in DragonDictate, Dragon's commercially available 30,000 word discrete utterance recognizer. In that system, the interface between the rapid match module and the recognizer is very straightforward. Each time the user says something, the rapid match algorithm provides the recognizer with a relatively short list of words that it thinks might have been said -- typically between 100 and 200 words -- and in fact, this list is usually supplied to the recognizer before the speaker has finished saying the word.

The nature of the interface between the rapid match module and the recognizer is different when continuous speech is involved, because the recognizer must contemplate hypotheses that represent sequences of words, in the course of recognizing an utterance. In the system we describe [1], the fundamental act that the rapid match module has been designed to carry out is to provide a short list of words that might begin at a particular time based on looking at speech data beginning at that time and extending a fixed (and short) duration into the future. Thus, whenever the recognizer is considering a partial sentence hypothesis that involves finishing a word at a certain time and needs to know what word hypotheses to consider as possible extensions, it can call the rapid match module to obtain a short list of plausible extensions.

The key ideas that the algorithm relies on are linear segmentation, smoothing, acoustic clustering, and word start groupings. In subsequent sections we shall elaborate on these ideas and explain their role in rapid match. We shall then report on some empirical results, having to do with a particular task that Dragon has chosen to use for development purposes: the dictation of mammography reports, using a vocabulary of 842 words.

Other rapid match algorithms that are quite different in character have also been described in the literature [2], [3], and [4].

2. Description of the Algorithm

We begin by describing the kind of data that the algorithm works with and then move on to describe the models to which the data are compared. Finally, we describe the actual way in which the computation is done.

We suppose that an utterance is converted by a front end to a sequence of k dimensional vectors, one per frame: $x_1, x_2, \ldots, x_n$. At any time (i.e. frame) t the rapid match module is capable of hypothesizing a short list of words that might begin at that time, based on looking at the vectors $x_t, x_{t+1}, \ldots, x_{t+w-1}$, where w is the window size. In our current implementation, a frame of 8 parameters is computed once every 20 milliseconds, and the window size is 12; thus the analysis is based on 240 milliseconds of speech data.

The algorithm begins with the computation of a sequence of (k dimensional) smooth frames $y_1, y_2, \ldots, y_s$, based on the x's in the window. Thus we have

$$y_1 = \sum_{i=0}^{b-1} a_i x_{t+i}$$

$$y_2 = \sum_{i=0}^{b-1} a_i x_{t+c+i}$$

$$y_3 = \sum_{i=0}^{b-1} a_i x_{t+2c+i}$$

etc.

where b is the smooth frame window width, the a's are the smoothing weights (and are assumed to sum to 1), c is the offset of successive smooth frame windows, and s is the number of smooth frames. A little thought reveals that w =

b + (s - 1)c. At the present time the smoothing weights are all equal, there are three smooth frames, each smooth frame is computed from a window encompassing four regular frames, and successive smooth frame windows are offset by 4 frames. Hence the smooth frame windows are nonoverlapping. In the DragonDictate isolated word rapid match we also make use of linear segmentation and smoothing. In that sytem, 5 smooth frames are computed using overlapping windows with nonuniform weights. We have not yet optimized the choice of the smoothing algorithm for continuous speech.

In this way we therefore convert 240 milliseconds of speech data into 3 smooth frames, or 24 numbers. The smoothing that is done is intended on the one hand to produce a very parsimonious representation of the speech data to make further processing very quick, and on the other, to represent the data in a way that is not too sensitive to variations in the duration of phonemes, thus obviating the need

A word start grouping (WSG) consists of a collection of words whose beginnings are acoustically similar. A word may appear in several different WSGs since, depending on the context in which the word finds itself, its acoustic realization may vary considerably. At the present time, we have at most 4 different word start groups for a given word, relating to whether or not the word emerges from silence or speech, and whether or not the word ends in silence or in more speech. It may prove to be desirable to expand the number of different possible representations of a word beyond this, to include prior phonetic context, but to do so might increase the necessary computation. The generation of the word start groups is automatic and relies on a specialized clustering algorithm. Here are some typical groups from the mammography vocabulary:

A. medial, medially, mediastinum, mediastinal, mediolateral, needle, needed
B. severe, suggest, suggested, severely, suggests, suggestive
C. dense, density, denser, densities

Each word start group (WSG) also consists of a sequence of acoustic models for smooth frames that might be generated from that group. More specifically a WSG is represented by a sequence of r probability distributions in k dimensional space, where r is no greater than s (the number of smooth frames computed): $f_1, f_2, \ldots, f_r$. For most WSG's, we would have r = s, but for WSG's that are to represent words that are so short that they may not last long enough for all s smooth frames to be computed, we allow r < s. For example, short function words like "of", "in", "the", etc. , when embedded in speech spoken continuously, often last less than 240 milliseconds.

Currently, we assume that each probability density f is the product of k univariate probability densities (i.e. we assume that the k elements of each smooth frame y are independent). Furthermore, we assume that each univariate density is a double exponential distribution (Laplacian). Thus, a single element of a smooth frame vector y is assumed to be a random variable with a probability density of the form

$$f(z) = \frac{1}{2\sigma} \exp(-|z - \mu|/\sigma)$$

where μ is the mean (or median) and σ is the mean absolute deviation (MAD).

If we wish to assess the evidence for whether the current sequence of smooth frames represents words in a particular WSG, we compute the score

$$S = \frac{1}{r} \sum_{i=1}^{r} -\log f_i(y_i)$$

which is the average negative log likelihood of the smooth frames evaluated using the probability model for the WSG. Let us suppose that there are M word start groups; then it would be necessary to compute a score for each of these: $S_1, S_2, \ldots, S_M$. A considerable computational saving can be achieved by having a particlar probability density f appear as part of the model for multiple different WSG's. Then, the very same value of -log f can be added into multiple different WSG scores. Obtaining a representation of the probability distributions of word start groups in terms of a small number of probability densities f is again a job for a specialized clustering algorithm, one that clusters probability distributions.

Once we have computed the scores for each of the WSG's, we throw out all those groups with scores worse than a threshold T_1. Then we look up all of the words contained in the surviving word start groups (throwing out any duplicates) and to each such word w, we attach the sum of its WSG score and a language model score:

$$S_w = S_{WSG} + S_{LM}$$

Finally we find all words w for which

$$S_w \leq T_2$$

where T_2 is a second (combined) threshold. At this point we have a list of words for the recognizer to contemplate in more detail. If the recognizer has asked us to return no more than p words, where p happens to be less than the number of survivors, we would prune the list further by throwing out the worst scoring candidates.

3. Some Results on the Mammography Task

The recognition task that has been used during the development of our rapid match algorithm has been an 842 word vocabulary drawn from mammography reports, many of which are actually transcriptions of oral dictations of radiologists. Since many of the words in the vocabulary can be pronounced in more than one way, we have added181 additional prounciations; thus the effective vocabulary size is 1023. With this vocabulary size, we have found that running our continuous speech recognizer with rapid match speeds up the system by a factor of 5 to 10, relative to running without it.

One obvious way of assessing the quality of the algorithm is simply to run the continuous speech recognizer with and without rapid match, and observe how many errors are introduced (or removed) by virtue of its use. On a test set of 1000 sentences,spoken by one person, consisting of a total of 8571 words, the word error rate was 3.7% running with rapid match returning a list of around 40 words per frame, and it was also 3.7% running without rapid match; in this particular experiment the total number of errors happened to be exactly the same, although the actual errors were somewhat different. This strategy for assessing performance is rather global , however, and it proved to be useful to have an assessment tool which provides more detail on where rapid match makes mistakes.

By running the recognizer in a mode where it know s the correct transcription of each of the utterances, it is possible to compute a reasonable segmentation of each utterance; that is to say, we can compute in which frame each word in the transcription is most likely to begin. Then, we can ask thisquestion for each word in each utterance: what is the rank of the score of the correct word among the candidates returned by the rapid match module in the frame in which the word begins (according to the segmenter)? At Dragon, there is an interactive program that has been written for the purpose of studying this question; by using it, the investigator finds it easy to detect words which have bad or inadequate rapid match models. Its basic functionality is to display and record the words that rapid match passes through to the recognizer in given frames. Table 1 displays the percentage of the time that the rapid match algorithm passes through the correct word in the correct frame as a function of the list size that the recognizer requests. Beyond a list size of 40, there are diminishing returns.

Size of list	Percentage passed through
10	84%
20	91%
30	94%
40	96%

Table 1. Percentage of words for which rapid match passes through the correct word in the frame in which the word begins, based on 700 tokens

One of the important features of the algorithm that the evaluation program highlights is that even if a word is not passed through in the "ideal" frame (ideal from the Hidden Markov Model's point of view), it is very likely to be passed through in a nearby frame. Because of the flexibility of dynamic programming, the inflexibility of our linear segmentation based rapid matcher does not prove to be as much of a disadvantage as one might have guessed. This observation is reinforced by the fact that even though 4% of the time the correct word does not appear in the rapid match list in the correct frame, running with the rapid match procedure (in a sufficiently conservative mode) produces no degradation in overall recognition performance, relative to running without it. In many cases where rapid match fails to pass back the word in the correct frame but does pass it through on a nearby frame, one sees upon inspecting a spectrogram of the utterance that the segmentation is hard to do by eye, and that the segmenter has made a reasonable choice for itself but not for the rapid matcher. In those cases the recognizer very often gets the word correct.

4. Conclusions

The rapid match procedure that we have described here appears to be a very promising method. On an approximately 1000 word continuous speech recognition task—a task that is not artificial, although another 1500 words would need to be added to give adequate coverage of real mammography reports in actual practice—it has enabled us to obtain near real time performance on a 33 megahertz 80486 processor.

We have done some preliminary work on adapting to new speakers based on several hundred mammography sentences, and we are optimistic that our rapid match models will adapt successfully, as they have inside of DragonDictate. A future paper will discuss the general question of the training and adaptation of these models.

REFERENCES

[1] P. Bamberg, Y. Chow, L. Gillick, R. Roth, D. Sturtevant, "The Dragon Continuous Speech Recognition System: A Real-Time Implementation", *Proceedings of DARPA Speech and Natural Language Workshop*, June 1990 Hidden Valley, Pennsylvania.

[2] Lalit Bahl, Raimo Bakis, Peter V. de Souza and Robert L. Mercer, "Obtaining Candidate Words by Polling in a Large Vocabulary Speech Recognition System", *ICASSP 88*, New York City, April 1988.

[3] Xavier L. Aubert, "Fast Look-Ahead Pruning Strategies in Continuous Speech Recognition", *ICASSP 89*, Glasgow, May 1989.

[4] Lalit Bahl, P. S. Gopalakrishnan, D. Kanevsky, D. Nahamoo, "Matrix Fast Match: A Fast Method for Identifying a Short List of Candidate Words for Decoding", *ICASSP 89*, Glasgow, May 1989.