Large Vocabulary Natural Language Continuous Speech Recognition*

L. R. Bakis, J. Bellegarda, P. F. Brown, D. Burshtein,
S. K. Das, P. V. ed Souza, P. S. Gopalakrishnan, F. Jelinck, D. Kanevsky,
R. L. Mercer, A. J. Nadas, D. Nahamoo, M. A. Picheny

The present paper describes our current research on automatic speech recognition of continuously read sentences from a naturally-occurring corpus: office correspondence. The recognition system combines features from our current isolated-word recognition system and from our previously developed continuous speech recognition systems. It consists of an acoustic processor, an acoustic channel model, a language model, and a linguistic decoder. Some new features in the recognizer relative to our isolated-word speech recognition system include the use of a fast match to rapidly prune to a manageable number the candidates considered by the detailed match, multiple pronunciations of all function words, and modelling of interphone coarticulatory behavior. To date, we have recorded training and test data from a set of 10 male talkers. The test data consist of 50 sentences drawn from spontaneously generated memos covered by a 5000 word vocabulary. The perplexity of the test sentences was found to be 93; none of the sentences were part of the data used to generate the language model. Preliminary (speaker-dependent) recognition results on these talkers yielded an average word error rate of 11.0%.

1. Introduction

The present paper describes our current research on automatic speech recognition of continuously read sentences from a naturally-occurring corpus: office correspondence. In previous work, we have concentrated on recognition of continuously read sentences from a 250 word vocabulary finite state grammar [1], continuously read sentences from a 1,000 word naturally-

[*] This is a paper originally presented at the Seoul International Conference on Natural Language Processing on November 22, 1990.

occurring corpus [1], and sentences from 5,000 and 20,000 word naturally-occurring corpora read with pauses between words [2,3]. This paper extends the previous work towards the recognition of continuously read sentences from a natural corpus covered by a 5,000 word vocabulary.

2. Task Description

The office correspondence task was developed by taking a large quantity of IBM internal electronic mail, determining the most frequently occurring 5, 000 words, and selecting from this database sentences fully covered by the 5,000 word vocabulary for test and training purposes [2].

For our experiments, we record a set of 10 male talkers reading training scripts of 2,000 sentences and several test scripts of varying size and recognition difficulties. This paper will report results on one of the test scripts (the "RSX" script) consisting of 50 sentences fully covered by our 5,000 word vocabulary. The training script consisted of sentences fully covered by a 20,000 word vocabulary; the first 500 sentences were the same for each talker, while the other 1,500 were different from talker to talker. The average sentence length for the training sentences was 16.4 words, and for the test sentences, 11.8 words. The talkers were from the local New York area. None were IBM employees. All recordings were made in a quiet office environment using a Crown PZM 6S microphone with a 12 bit analog to digital (A/D) converter. The range of the talker's speaking rates was broad; the fastest talker spoke at 170 words per minute (wpm) and the slowest, at 130 wpm. It took each talker approximately one week to record the necessary speech.

3. Description of the Base Recognition System

The recognition system is based on features present in our current isolated-word recognition system and in our previously developed continuous speech recognition system. It consists of an acoustic processor, an acoustic channel model, a fast matcher, a language model, and a hypothesis search module. Thus the overall configuration is that described in reference [4].

The acoustic processor extracts a vector of 20 spectral features from the speech signal, and quantizes each feature vector into one of 200 possible prototype classes. The acoustic channel model describes in a probabilistic fashion the way in which words are realized as sequences of prototypes produced by the acoustic processor. The fast matcher produces a short list of words whose uttering could have caused an indicated acoustic processor prototype string. The language model estimates the probability of the next word in the sentence given the previously hypothesized words in the sentence. The hypothesis search module directs the recognition process, maintaining a tree of currently active hypothesized subsentence paths. It evaluates their likelihood and, accordingly, discards some paths and extends others.

There are several modifications that we made to the first two components of our basic recognition system in order to obtain improved continuous speech recognition performance. These will be described in the following sections. This section sketches the basic system.

The spectral feature extraction of our acoustic processor is based on an adaptive auditory("ear") model described by Cohen [5]. The processor determines the Euclidean distance of each feature vector produced by the ear model to all 200 prototype vectors, and puts out the identifier(label) of the nearest prototype.

In isolated word experiments Markov model for a word was determined from 10 utterances of the word from 10 speakers, and did not depend on the context [6]. However, to accommodate the requirements of continuous speech, the model for any particular word depends on the immediate word context. The principles of this dependence are outlined in the next section.

As recognition proceeds, the hypothesis search module endeavors to extend particular hypothesized paths specified by a sequence of words (or rather lexemes – see the next section) starting with the beginning of the current sentence (our search units are words/lexemes). Since in a natural text task, every word can be followed by any other one with a varying but non-zero probability, the fit of all the words of the vocabulary to the unaccounted-for portion of the acoustic processor output string should be examined. Since the vocabulary is large, the examination is carried out in two steps. The first step, called the Fast Match, reduces the possibilities to a few candidates (30 on average) whose fit, relative to an acoustic Markov model, is then evaluated by the Detailed Match. One version of the Fast Match is described in reference [7]. The current recognizer organizes its Fast Match around a tree constructed from the phonetic baseforms [4,6] corresponding to the words of the vocabulary. The branches of the tree are Hidden Markov Models(HMMs) determind by the phone in question. These component HMMs are of a simplified variety (the distributions of all the transitions are identical) allowing fast computation. Thresholding is used to prune this tree by eliminating those paths that do not fit the acoustic label sequence submitted to the Fast Match. The resulting shorter list of acoustically compatible candidate words is further pruned by the language model that eliminates some of the a priori less likely continuations of the hypothesized path being extended.

The recognizer uses our standard trigram language model [1] which is based on an interpolation of relative frequencies of trigrams, bigrams, and unigrams collected from a 200 million word text data base. The interpolation weights are determined by the method of deleted interpolation [1,Section VIII]. The n-igrams used(n=1,2,3) are sequences of *n* consecutive words in the training corpus that belong to the basic vocabulary.

The hypothesis search is, in principle, that described in Section VI of reference [1], and is based on the stack algorithm of sequential decoding [8]. The acoustic component of the likelihood score is provided by the acoustic model (see next section), and its linguistic component by the trigram language model. However, path extensions are carried out only for the words specified by the Fast Match component.

4. A Contextual Allophonic Acoustic Model

The basic principle of our acoustic model is as follows. To each word of the vocabulary there correspond one of more basic pronunciations, called lexemes. There is also a silence lexeme. The pronunciation of a lexeme is specified by a *baseform*, which is a sequence of symbols from a phonetic alphabet of size 64. For instance, the word ⟨either⟩ corresponds to two lexemes with baseforms *ee1 dh er0* and *ai1 ixg dh er0*, respectively. Our recognizer actually decodes sequences of lexemes rather than words.

Each of the 64 phones (phonetic symbols) F can realized by a variety of allophones F(1), F(2), ⋯, F(K), and so a baseform B_1, B_2, ⋯, B_n is realized by an allophonic sequence $B_1(i_1)$, $B_2(i_2)$, ⋯, $B_n(i_n)$ ($B_j(i_j)$ denotes the i_j allophone of the j^{th} phone of the lexeme) whose identity is determined by the phones of the lexeme and of the hypothesized lexeme string being extended (between the lexeme and the preceding path there is always inserted a word separation phone). The variant i_j of phone B_j depends on the class identity of a string of phones centered by B_j. The equivalence classification is determined by use of decision trees [9] and depends on pre-training data, as does the variety of allophones of each phone.

To each allophone $F(i)$ there corresponds a Markov model, and thus the baseform is the concatenation of the Markov models corresponding to the allophones whose string realizes the lexeme. This baseform them determines the acoustic model of the lexeme in the particular context of the neighboring lexeme string.

Each transition in any of the Markov models is identified with one of the arcs in an inventory of 200 arcs. Transitions identified with the same are restricted to have the same output probability distribution over the 200 acoustic processor labels.

5. Supervised Vector Quantization

The 200 prototype vectors used by the acoustic processor are selected in an iterative mode intended to optimize the efficiency of the allophonic acoustic model. The procedure is based on the intuitive notion that the individual prototypes should represent the individual arcs in the arc inventory (there is an equal number, 200, of arcs and prototypes), because the latter are the phonetic means used to describe pronunciation.

We proceed as follows. We obtain original prototypes by "ordinary" vector quantization. Since the training script determines a lexeme string which in turn models, then determines an allophone string, and each allophone corresponds to a Markov model, then the training feature vector string produced inside the automatic processor [Section 3] corresponds to a particular sequence. Using allophonic Markov models (whose statistics are determined by forward/backward training), we can Viterbi align the feature vectors and arcs of the allophone models. For each arc in the arc inventory we then assemble a collection of feature vectors aligned with it. The 200 collections then lead to a new set of prototypes. This set is the basis of the next iteration of the process : re-labeling of acoustic processor output; determination of the allophonic varieties of all phones, and of phone string equivalence classification determining the allophone string realization of lexemes; estimation of acoustic model statistical parameters; alignment of feature vectors and model arcs; and creation of the next generation of prototypes. Iteration continues as long as a perceptible change in the prototypes is observed.

6. Some Additional Recognizer Adjustments

Many of the very frequent words (we call them arbitrarily, *function words)* are short and are, in continuous speech, carelessly pronounced, so they can benefit by careful treatment [10]. There being only 130 such words, we can afford to model them as individual special phones. This can easily be accommodated in the framework of the contextual allophonic acoustic model of Section 4.

Speakers sometimes pause at appropriate points in a sentence. The hypothesis search module provides for this possibility by allowing the extension of a path by a silence lexeme. The trigram language model skips this lexeme in the path history when predicting the next word.

The hypothesis search determines the match score for a word by dividing the actual probability for the word as computed by the model by the expected value of the match score for the word, given the correct word model. A bias that increases linearly over time is added to force the match score to tend to increase over time on the correct path. The match is terminated when the correct score falls below a preset threshold. In isolated-word speech, the bias term can be set quite high, as the silence that occurs after each word will always allow us to determine when to terminate the match. In continuous speech, a high bias term causes the match to continue over short words, e.g., "*do you want us*" is recognized as "*he was*", while a low bias term tends to break long words into short ones. We found that a much smaller bias than used in isolated word speech produced much better performance in continuous speech.

7. Training

It was mentioned in Section 2 that ten(10) talkers read 2,000 sentences. The totality of this data is used to determine (in pre-training) the allophonic variety for the phone set, as well as the equivalence classification determining the desirable allophone from the context preceding and following phones (see Section 4). This specifies the lexeme to allophonic correspondence. The statistical parameters of the HMMs are then determined for the speech of the individual speaker to be recognized.

The training will result in proper estimation only if based on the correct lexeme (rather than word) script. The speakers are given an ordinary text to read without being instructed where to pause or how to pronounce each word. Their speech must thus first be subjected to a decision process which determines the location of pauses as well as the identity of the speakers' choices in multi-lexemic words.

8. System Performance

For comparison, we will give the result of four(4) experiments dealing with recognition of the natural text covered by a 5,000 word vocabulary: isolated word recognition with context-independent phonetic and fenonic [6] models, and continuous speech recognition with context-independent and allophonic models. The system was trained on all 2,000 sentences from each talker; for isolated word speech, only 100 sentences were available for training, the test script was the "RSX" script (above). Only error rates computed for word deletions and substitutions are reported [10].

Table 1 — Test Results on RSX script under various conditions.

	Isolated Speech		Continuous	Speech
	Phonetic	Fenonic	Phonetic	Allophonic
	(%)			
T1	5.6	2.4	25.0	8.8
T2	4.7	3.2	33.1	13.0
T3	5.4	3.7	39.7	18.2
T4	6.3	4.6	28.7	13.0
T5	2.0	1.5	11.7	6.1
T6	7.1	3.0	42.1	13.2
T7	3.5	1.5	24.3	11.5
T8	4.1	1.9	13.4	6.3
T9	5.6	2.5	22.0	8.5
T10	16.2	8.5	28.1	12.2
AVG	6.1	3.3	26.8	11.0

Approximately 1/3 of the errors for the allophonic models are caused by the fast match not returning the correct lexeme to be processed by the detailed match. Approximately 2.5 CPU hours on a large IBM mainframe is required per talker for continuous speech recognition. This is more than 25 times the CPU time needed for the isolated word task; this figure does not include signal processing, training, and supervision time. Note that the allophonic models produce a larger performance gain relative to context-independent phone models in continuous speech than the fenonic models in isolated word speech. Some of the additional performance gain may be attributable to the use of supervised vector quantization; this was not pursued in the isolated word experiment because of a lack of training data. Future work will include exploring new fast match strategies, better labelling methods, and comparisons to other techniques for performing context-dependent modeling [10].

Continuous Speech Recognition Group
Computer Science Department
—IBM Research Division
Thomas J. Watson Research Center
P. O. Box 218, Yorktown Heights, NY 10598
U. S. A.