ELSEVIER

Speech Communication 14 (1994) 49–60

SPEECH COMMUNICATION

Experiments in continuous speech recognition using books on tape *†

P. Kenny *, G. Boulianne, H. Garudadri, S. Trudelle, R. Hollan †, M. Lennig †, D. O'Shaughnessy

INRS-Télécommunications, 16, place du Commerce, Verdun (Ile-des-Soeurs), Québec, Canada H3E 1H6

(Received 25 January 1993; revised 20 August 1993)

Abstract

We present a new search algorithm for very large vocabulary continuous speech recognition. Continuous speech recognition with this algorithm is only about 10 times more computationally expensive than isolated word recognition. We report preliminary recognition results obtained by testing our recognizer on books on tape using a 60 000 word dictionary.

Zusammenfassung

Wir stellen einen neuen Suchalgorithmus zur Erkennung von Kontinuierlicher Sprache bei sehr großem Wortschatz (50 000–100 000 Worte) vor. Mit diesem Algorithmus benötigt man für die Erkennung kontinuierlicher Sprache nur das Zehnfache der Rechenleistung, die zur Erkennung von isolierten Worten benötigt wird. Wir berichten über unsere vorläufigen Erkennungsresultate, die wir durch Tests unseres Programms an einem Wortschatz von 60 000 Worten erhalten haben. Dieser Wortschatz stammt aus auf Tonbänder aufgenommenen Büchern.

Résumé

On présente un nouvel algorithme de recherche pour la reconnaissance de la parole continue à très grand vocabulaire. La complexité de cet algorithme augmente de seulement dix fois en passant des mots isolés à la parole continue. On donne des résultats préliminaires obtenus en testant le système de reconnaissance sur des livres sur cassette utilisant un dictionnaire de 60 000 mots.

Key words: Continuous speech recognition; A* search; Hidden Markov model

1. Introduction

In this paper we will report our initial efforts to extend our earlier work on very large vocabu-

* Corresponding author.

*† This work was supported by the Natural Sciences and Engineering Research Council of Canada.

† Also with Bell-Northern Research, Montréal, Canada.

SSDI 0167-6393(93)E0078-A

lary isolated word recognition (Kenny et al., 1993a; Deng et al., 1991; Lennig et al., 1990) to continuous speech. This project is intended as an exploratory study of the feasibility of domain-independent continuous speech recognition and it is not geared to an immediate commercial application.

In order to develop our continuous speech training and recognition algorithms we decided to work with commercially distributed books on tape (analog cassettes). In fact we had little choice in the matter since, when we began work, the *Wall Street Journal* Corpus had not yet been collected and books on tape was the only abundant source of transcribed speech data which was readily available to us. Since this data is not segmented into sentences (unlike the *Wall Street Journal* Corpus) we had to design algorithms for training and recognition which are capable of handling unsegmented data files of arbitrary length. We hope that our approach will be of interest to researchers working in languages other than English (or dialects of English other than General American) who do not have large corpora of segmented speech data available to them. For our part, we have recently begun experimenting with books on tape recorded by speakers of Quebec French.

We will report results of experiments performed on six books recorded by three male and three female speakers. We took care to choose unabridged recordings without sound effects and we used an optical character recognizer to read the accompanying texts. Errors made by the optical character recognizer were corrected manually and the texts were adjusted to account for errors made by the speakers. For each book, we designated the first and last third of each chapter as training data and the middle third as test data. In each case, we used 1–2 hours of training data to build a collection of speaker-dependent acoustic phonetic models and performed recognition experiments on a 500–1000 word subset of the test data. Although all of the experiments reported here were performed under conditions of speaker-dependence, we believe that our approach can easily be extended to the speaker-independent case. We do not know how many books would have to be used to construct adequate speaker-independent models, but there is evidence that a relatively small number would probably be sufficient. Kubala and Schwartz (1990) report that recognition results on the DARPA Resource Management task obtained with a relatively large amount of training data collected from only 12 speakers (7 male and 5 female) were comparable to their best results obtained with training data collected from 109 speakers.

The books we choose are equally divided between fiction (a novel by Jack London and two by Henry James) and non-fiction (Helen Keller's autobiography and two books on current affairs). In order to perform recognition experiments with such diverse material we had to make some decisions concerning the dictionary and language model. We were confronted with the same issues in designing our experimental isolated word recognizer. In that case, we found that we could obtain an average recognition rate of 93% on a variety of texts drawn from newspapers, magazines and novels by configuring the dictionary and language model so as to be domain-independent. We used a dictionary containing transcriptions of 86 000 words (Seitz et al., 1990) which was not tailored to any of the test sets (although care was taken to ensure that all of the words in the test sets were contained in the dictionary). Similarly we used a language model trained on 60 million words of newspaper text (Gupta et al., 1992a,b) which was not adapted to any of the test material. This language model assigns a "wild-card" score to words in the dictionary that are not covered in the language model training data, so no special provision was needed to handle such words when they were encountered in recognition (but naturally they were a very frequent source of errors (Gupta et al., 1992a,b)).

We used nine speakers to test the isolated word recognizer. Five of the speakers read newspaper and magazine articles in isolated word mode for training and testing. The remaining four speakers read from novels (namely *Tender is the Night*, *A Confederacy of Dunces*, *Slaves of*

New York and *Neighbors*). It was therefore quite natural for us to use books on tape as training test material for developping our continuous speech algorithms and to adopt a similar domain-independent approach in designing our recognition experiments. We found that a dictionary of 60 000 words (obtained by removing the very infrequent words from the 86 000 word dictionary) contained more than 98% of the words in the six books we were using. So we decided to add the missing words from each of the books and use the same vocabulary in all of our experiments. Similarly we decided to use a common language model, namely the newspaper trigram model referred to above, at least for our first series of experiments. This language model was used in exactly the same way as in the isolated word recognition experiments. We did not adapt the language model to the individual books and words in the dictionary which are not covered in the language model training data were treated simply as wildcards. (But see (Zhao et al., 1993) for some preliminary results on language model adaptation.)

We obtained an average recognition accuracy of 72% under these conditions (detailed results are given in Section 4). This is rather low when compared with the figure of 93% that we obtained in our isolated word recognition experiments, but it is not surprising in view of the difficulty of the task. We believe that we will make more progress in the long run by tackling difficult tasks such as this than by limiting ourselves to tasks where we can be confident of obtaining respectable recognition accuracies from the outset. Our work on isolated word recognition with an 86 000 word vocabulary was conducted in this spirit and it lead to the development of the very successful STOCKTALK application by the Montreal laboratory of Bell-Northern Research (Lennig et al., 1992). Similarly, we found that by tackling the problem of continuous speech recognition with a 60 000 word dictionary (containing 130 000 phonemic transcriptions) we were forced to develop new search techniques (Kenny et al., 1993a) which found immediate application in the STOCKTALK system.

2. The search strategy

In order to perform our recognition experiments, we have developed a new approach to the search problem which extends our earlier work on searching in the context of isolated word recognition (Kenny et al., 1993a). The major distinguishing features of our isolated word recognition algorithm are

(i) It is phone-synchronous rather than frame-synchronous. (The search advances one phone at a time rather than one frame at a time.)

(ii) It is bi-directional (Soong and Huang, 1991; Zue et al., 1991; Austin et al., 1991; Kenny et al., 1993a). More precisely, it is an A* search (Nilsson, 1982) guided by a heuristic obtained by a reverse-time search of a phonetic graph which imposes triphone phonotactic constraints on phoneme strings.

(iii) The heuristic is used to identify the end time of the third-to-last phoneme in each partial recognition hypothesis (using the "2-phone lookahead" property (Kenny et al., 1993a)).

(iv) The acoustic matches of every segment of data with each of the phoneme models (the "point scores" (Kenny et al., 1993a)) are precomputed before carrying out the search. In our current work, phonemes are modelled with HMMs and the acoustic match of a segment with a phoneme model is calculated using the Viterbi algorithm. However, the algorithm can incorporate *any* mechanism for calculating point scores; in particular non-Markovian models of segment-level features such as energy and duration (Kenny et al., 1991; 1993a; Sagayama, 1991) can be combined with Markovian spectral models (acoustic HMMs).

When applied to speaker-dependent isolated word recognition with a 60 000 word vocabulary, most of the computation is taken up by the preprocessing (the calculation of the point scores and the Viterbi search needed to evaluate the heuristic) and the A* search itself accounts for only about 1% of the total. In our continuous speech recognition experiments (which were per-

formed under essentially the same conditions, the only major difference being that the quality of the audio signal was not as good in the case of the books on tape) we have found that the amount of pre-processing per unit time remains roughly the same, but the amount of computation needed for the A* search increases by three orders of magnitude. Hence, *the total computational demands of the algorithm only increase by a factor of about 10.*

In extending the isolated word recognition algorithm to continuous speech we decided to continue to use the phoneme as the fundamental search unit rather than the word. Such an approach to continuous speech recognition has been extensively developped by Ney et al. (1992) on a 10 000 word vocabulary in German. Their decoding algorithm consists of a Viterbi search of the hidden Markov model obtained by combining phoneme HMMs with a Markovian language model (such as a trigram model). Aggressive pruning is necessary since the search space is very large. If a bigram language model is used, then the search space contains one copy of the lexical tree for every word in the vocabulary and in the case of a trigram language model, a copy of the lexical tree is needed for every possible bigram. (But see (Austin et al., 1990; Paesler and Ney, 1989) for methods of reducing the effective size of the search space.)

The alternative word-based approach to the search problem is usually based on stack decoding (Jelinek, 1969, 1976; Paul, 1991). It depends on having a good fast match strategy to identify candidate words whenever a word boundary is hypothesized. Many different fast match algorithms have been proposed (Bahl et al., 1988, 1993; Fissore et al., 1989; Gupta et al., 1988), but they have yet to be shown to perform satisfactorily on continuous speech tasks having very large vocabularies. However our principal reason for rejecting the word-based approach in favour of the phoneme-based approach (at least in our initial experiments) is that the phoneme-based approach is computationally more efficient in that it does not require that the lexical tree be searched exhaustively every time a word boundary is hypothesized. The major drawback of the phoneme based approach is in terms of memory requirements, since many copies of the lexical tree have to be searched simultaneously. The memory requirements per unit time of our phone synchronous approach are in any case much greater than those of the classical frame synchronous approach, so it was clear to us at the outset that we could not attempt to recognize an entire sentence at a time. Rather, we had to devise a strategy to break the speech data in a file into blocks of reasonable size and process the data one block at a time. (We encountered a similar problem in training since our training data was not segmented into sentences. Our approach to the training problem is described in the companion paper (Boulianne et al., 1994).)

Aside from the need to control memory usage, there is a more fundamental reason why block processing is necessary. It is obviously not possible to recognize speech in real-time using a bi-directional search algorithm which requires that the second pass through the data be postponed until the first pass has been completed. The only scenario in which a bi-directional search strategy can achieve real-time is by processing the data in blocks in such a way that the two passes through the data in a given block are carried out while the data in the next block is being captured. (To be precise, the scenario we are describing is real-time with a lag of one block rather than strict real-time.)

Our overall strategy can be summarized as follows. (Details of the computation needed to recognize the data in a given block are given in the next section.) We break the data file to be recognized into blocks of equal length and we use an A* search in each block which is similar to the isolated word recognition algorithm except insofar as word boundaries are not known in advance. As in the isolated word case, we construct an admissible heuristic by means of an initial Viterbi search through a graph which imposes triphone phonotactic constraints on phone strings. The A* search generates a list of theories (partial phonemic transcriptions together with word histories) for the speech data up to the end of the block [1]. As soon as the list of theories for the current

block has been obtained, the block is swapped out of memory and the search of the next block begins using this list to initialize the stack.

This list of theories plays the same role as the beam used in a time synchronous Viterbi search. The Markov property of the trigram language model allows us to merge theories that have identical recent pasts but different remote pasts. (The stack decoder described by Paul (1991) also takes advantage of this fact.) When this merging is carried out, the number of theories that have to be generated at the end of each block (the "beam width") can be held fixed without running the risk of losing the optimal theory. In order to pursue the search in subsequent blocks, the only information needed concerns the recent pasts of these theories. By logging the information concerning the remote pasts to disk we are able to ensure that the memory required to recognize a file is independent of its length (instead of increasing exponentially with the length of the file as would be necessary without merging and block processing).

Finally, by tracing back the highest scoring theory on the beam after the last block has been processed, we obtain the recognition hypothesis which best accounts for all of the data in the file. The recognition algorithm can therefore be viewed globally as a beam search and locally as an A* search.

The experiments reported here have been conducted using context-independent phoneme models and allophone models (Bahl et al., 1991) defined by contexts which do not extend across word boundaries. Although the search algorithm can be extended to accommodate cross-word allophone models fairly easily, our experience leads us to believe that if the number of allophone models is large (say several thousand), then considerations of efficiency will require a two-pass approach in which detailed allophone models are not used in searching but only in rescoring hypotheses returned by the search.

3. Searching a block

Suppose we are given a block of data $[T_1, T_2]$ (the unit of time is the frame and the frame advance is 10 ms). We search the data in the block using a stack decoder which proceeds as follows. At each iteration of the search, there is a sorted list (or "stack") of theories each with a heuristic score. This heuristic score is a combination of two scores, one calculated in the forward direction and the other calculated in the backward direction. The forward score is the exact likelihood of the speech data accounted for by the theory (calculated using acoustic HMMs and the language model) and the backward score is an overestimate of the likelihood of the remaining data on the optimal extension of the theory permitted by the lexicon and the language model. The theory with the highest heuristic score is expanded, meaning that, for each of the one-phoneme extensions permitted by the lexicon, the heuristic score of the extended theory is calculated and the extended theory is inserted into the stack at the appropriate position. This process is iterated until sufficiently many theories satisfying a suitable termination criterion have been generated.

We have to explain how the backward scores are calculated, what data structure used to represent a theory, how the stack is initialized prior to searching the block, what criterion is used to determine when a theory is complete and, finally, how theories having identical recent pasts but different remote pasts can be merged so as to speed up the search.

3.1. Calculating the backward scores

Our strategy for calculating the backward scores is essentially the same as in the isolated word case, that is, we conduct an exhaustive search in the reverse time direction through a

[1] More precisely, each of the theories generated has the property that all of the hypothesized end times for the third-to-last phoneme in the partial phonemic transcription are beyond the end of the block. The partial phonemic transcription need not end at a word boundary.

phonetic graph which imposes triphone phonotactic constraints on phoneme strings. This graph is specified as follows and is denoted by G^*.

(i) *Nodes*: there is one node for every possible diphone fg.

(ii) *Branches*: for every legitimate triphone fgh (that is, a triphone that can be obtained by concatenating the phonemic transcriptions of words in the dictionary) there is a branch from the node corresponding to the diphone fg to the node corresponding to the diphone gh.

(iii) *Branch labels*: if fgh is a legitimate triphone then the branch from the node fg to the node gh carries the label f.

We construct a hidden Markov model M by replacing each of the branches in this graph by a phonetic HMM. A detailed construction which avoids the use of null-transitions and encodes each node in G^* as a state in M is given in (Kenny et al., 1993a).

The backward scores used in searching the block $[T_1, T_2]$ are obtained by performing a Viterbi search through M in the reverse time direction, starting at time $T_2 + \Delta$ where Δ is a suitably chosen positive integer (the criterion used to choose Δ is given below). We have no a priori knowledge of what state in M is occupied at time $T_2 + \Delta$, so we initialize the backward recursion by setting equal to 1 the backward scores associated with each state in M at this time.

This backward pass gives, for each node n in G^* and each time $t = T_1 - 1, \ldots, T_2 + \Delta - 1$, the Viterbi score of the data in the interval $[t+1, T_2 + \Delta]$ on the best path in G^* which leaves n at time t and is subject to no constraints on the state in the model occupied at time $T_2 + \Delta$. We denote this quantity by $\beta_t^*(n)$.

Suppose we are given a partial phonemic transcription $f_1 \cdots f_k$. Let n be the node in G^* corresponding to the diphone $f_{k-1}f_k$ and for each time t, let $\alpha_t(f_1 \cdots f_{k-2})$ denote the Viterbi score of all of the data up to time t (starting from the beginning of the utterance) for the truncated transcription $f_1 \cdots f_{k-2}$. It is reasonable to estimate the end time of the phoneme f_{k-2} as $\operatorname{argmax}_t \alpha_t(f_1 \cdots f_{k-2})\beta_t^*(n)$. This is because $\beta_t^*(n)$ is the Viterbi score of the data in the interval $[t+1, T_2 + \Delta]$ on the best path in G^* which leaves n at time t and the graph G^* is constructed in such a way that this path is constrained to pass first through a branch labelled f_{k-1} and then through a branch labelled f_k.

In the case of clean speech and speaker-dependent models, this way of estimating end times turns out to be exact almost all of the time (the "2-phone lookahead property") but it is safer to hypothesize several end times (for instance by taking the five values of t for which $\alpha_t(f_1 \cdots f_{k-2})\beta_t^*(n)$ is largest).

3.2. Partial recognition hypotheses

A partial recognition hypothesis (or "theory") θ is a septuple $(\boldsymbol{w}, f, m, n, \sigma, \{\alpha_t\}, S)$, where

1. $\boldsymbol{w} = w_1 \cdots w_N$ is a word history;
2. $f = f_1 \cdots f_k$ is a partial phonemic transcription which may extend into a word following w_N (but there are no complete words after w_N in the partial transcription f);
3. m is a node in the lexical tree (Kenny et al., 1993a) corresponding to the part f which extends beyond w_N, if any; m is the root node of the lexical tree otherwise;
4. n is the node in the graph G^* which corresponds to the diphone $f_{k-1}f_k$;
5. σ is the current state of the trigram language model; there are three possibilities depending on whether the word following w_N is predicted using a trigram distribution $P(\cdot \mid w_{N-1}w_N)$, a bigram distribution $P(\cdot \mid w_N)$ or a unigram distribution $P(\cdot)$;
6. for each endpoint hypothesis t, α_t is the Viterbi score of the data up to time t against the model for the truncated transcription $f_1 \cdots f_{k-2}$;
7. S is the heuristic score which is given by

$$S = P(\boldsymbol{w}) \max_t \alpha_t(f_1 \cdots f_{k-2})\beta_t^*(n),$$

where $P(\boldsymbol{w})$ is the probability of the word string $\boldsymbol{w}$ calculated using the trigram language model.

The reason why both $\boldsymbol{w}$ and f have to be specified is that different words may have the same transcription and different transcriptions

may correspond to the same word. Obviously it is redundant to specify m, n and σ in addition to w and f but it is convenient to do so.

A stack entry is said to be *complete* if all of its hypothesized endpoints are to the right of T_2. The parameter Δ is determined empirically by the condition that the exact endpoint of a complete stack entry should always be included among the hypothesized endpoints. (Since it is not actually possible because of memory limitations to carry around sufficient information with each theory to be able to generate its segmentation, we test this condition by verifying that the acoustic score of the global transcription found by the recognizer of the data in each file is the same as the score found by the training program when it is run with this transcription.)

At the start of the search, the stack is initialized using the list of theories generated by searching the previous block (ending at time T_1). Each of these has the property that all of its hypothesised endpoints are to the right of T_1, so the speech data prior to the beginning of the current block is no longer needed. The search terminates when sufficiently many complete theories have been generated at which point the next block is swapped into memory and a new search begins.

The reader may have noticed that the forward and backward components of the score S are asymmetrical in that the forward component contains both a language model score and an acoustic score, whereas the backward component contains only an acoustic score. The requirement for admissibility (Nilsson, 1982) of a forward–backward heuristic scoring function is that the backward score associated with a theory dominate the forward score of any permissible extension of the theory, and the tighter the estimate provided by the backward score the more efficient the search will be.

To see that the admissibility condition is satisfied by the scoring function we have specified, fix a theory $(w, f, m, n, \sigma, \{\alpha_t\}, S)$ and observe that

(i) The graph G^* allows for more freedom in extending partial transcriptions than do the lexicon and language model, so, for every time t the backward score $\beta_t^*(n)$ dominates the acoustic score of the data starting at time $t+1$ for any partial transcription that begins with $f_{k-1}f_k$.

(ii) If w' is any word string that extends w, then $P(w' \mid w) \leqslant 1$ (since the language model is a discrete probability distribution).

It follows that the combined acoustic and language model score of any extension of the theory is dominated by S, as required.

The question naturally arises whether it is possible to construct a tighter bound on language model scores than that given in (ii). It is easy to build a phone-level language model which could be incorporated into the calculation of the β^*'s (using, say, statistics of triphone occurences derived from the word-level language model statistics) but it is not obvious how such a phone-level language model could be used to estimate the language model score of extensions of a given word string in an A*-admissible way. For this reason, we decided not to use a phone-level language model in the calculation of the β^*'s, at least in our initial experiments. However, the question of whether to use such a phone level language model is probably worth looking into, since the issue of admissibility may not prove to be important in practice. Admissibility guarantees that when a stack decoder is being used to search a graph, then the first N complete paths to appear on the top of the stack are precisely the N best paths in the graph. With a heuristic that is inadmissible but still reasonably accurate, it may be necessary to wait until more than N complete paths have appeared on the top of the stack in order to be sure of capturing the N best complete paths, but this is not a practical issue if the search is capable of finding complete paths quickly.

3.3. Merging

The Markov property of the trigram language model enables us to merge theories that have identical recent pasts but different remote pasts. Specifically, suppose we have two theories $\theta = (w, f, m, n, \sigma, \{\alpha_t\}, S)$ and $\theta' = (w', f', m', n', \sigma', \{\alpha_t'\}, S')$ such that $m = m'$, $n = n'$ and $\sigma = \sigma'$. (In this case we will say that θ and θ' are equivalent.)

The future extensions of both theories which best account for the data starting at any given time (subject to lexical and language model constraints) will be identical. Thus if it happens that t is on the list of hypothesized endpoints for both theories and

$$P(w')\alpha'_t < P(w)\alpha_t,$$

then we can remove t from the hypothesis list for the second theory without running the risk of losing the optimal path. In practice, the condition $n = n'$ means that the list of hypothesized endpoints for both theories will be the same (except in very rare cases). Furthermore, if this inequality holds for one such t, then it is typically because the first theory gives a better fit to the remote past than the second theory; hence it will usually be the case that if the inequality holds for one t, then it will hold for all t and the second theory can be pruned away completely.

We can take advantage of this fact to speed up the A* search by maintaining a list of "merge buckets" consisting of all the equivalence classes of theories encountered in the course of the search. Associated with each equivalence class we have an array of forward scores $\{A_t\}$ which is updated throughout the search. For each t, A_t is defined to be $\max_\theta P(w)\alpha_t$, where θ extends over all theories $(w, f, m, n, \sigma, \{\alpha_t\}, S)$ in the given equivalence class that have been encountered so far (in the course of searching the current block). When a new theory $\theta' = (w', f', m', n', \sigma', \{\alpha'_t\}, S')$ in this equivalence class comes to be inserted into the stack, for each hypothesized endpoint t, we can test to see if the inequality

$$P(w')\alpha'_t < A_t$$

holds. If it does, then we can prune this endpoint hypothesis before entering the theory into the stack; if not, then A_t is updated and the endpoint hypothesis has to be retained.

We have not been able to implement this scheme fully because of memory limitations. In practice, we only invoke merging when a word boundary is hypothesized, so the only merge buckets generated in the course of the search are those which correspond to theories for which m is the root node of the lexical tree. (However, before starting the search we prune the list of hypotheses generated by searching the previous block by merging at arbitrary phoneme boundaries and we use this pruned list to initialize the stack.)

4. Experimental results

The books on tape that we have been using for development purposes consist of three novels, namely *White Fang*, *The Europeans* and *Washington Square* and three non-fiction works, namely Helen Keller's autobiography, and two collections of essays dealing with current affairs, namely *Preferential Policies* and *All It Takes Is Guts*.

In the recognition experiments we used a dictionary of 60 000 words, containing an average of 2.2 transcriptions per word, which was augmented to include all of the words all of the books. (About 1.5% of the words were missing in each case. The majority of the missing words were inflected forms of words already in the dictionary.) The language model used was a trigram language model trained on 60 000 000 words of newspaper text (Gupta et al., 1992a). As in our isolated-word recognition experiments we used this language model a "black box" without adapting it to any of the test domains.

The speech data was digitized at 16 kHz and the acoustic features used for our experiments consisted of a set of eight static and seven dynamic mel-based cepstral coefficents (calculated every 10 ms). We trained context-indendent (CI) and context-dependent (CD) phonetic HMMs for each of the speakers using the algorithm described in the companion paper (Boulianne et al., 1994). The context-independent models consisted of a collection of 41 mixture HMMs (one model per phoneme). The output distributions in each of the models were 25-component Gaussian mixtures and they were associated with the transitions in the model rather than states. For each model, we used a single covariance matrix common to all mixture components in all of the transitions in the model. The context-dependent models consisted of 137 mixture HMMs of the same type with covariance matrices tied among

Table 1
Training and test set sizes and recognition results for six books

Book (Sex)	Training	Test	Accuracy (CI)	Accuracy (CD)
White Fang (M)	13,200	730	58%	68%
Washington Square (F)	15,486	586	75%	79%
The Europeans (F)	19,158	499	64%	69%
Helen Keller (F)	19,811	1005	66%	71%
Preferential Policies (M)	21,938	696	70%	75%
All It Takes is Guts (M)	19,311	561	67%	71%

the allophones of each phoneme. Allophone clustering was performed by means of a decision tree (Bahl et al., 1991) but no attempt was made to model cross-word effects. Our principal results are shown in Table 1.

The second and third columns in this table give the training and test set sizes in words; the fourth and fifth columns give the recognition accuracies for experiments performed with context-independent (CI) and context-dependent (CD) HMMs.

The accuracies were calculated using the formula

$$\frac{1}{N}\left(N - \left(\text{Substitutions} + \tfrac{1}{2}[\text{Deletions} + \text{Insertions}]\right)\right),$$

where N is the size of the test set. This way of evaluating the accuracy is unusual in that substitutions, deletions and insertions are not assigned equal penalties. The reason for the factor of $\frac{1}{2}$ is to ensure that the formula for evaluating the accuracy gives the same weight to a substitution as to a combination of a deletion plus an insertion. In the case of context-independent models, the substitution rate (averaged across all books) was 27.2%, the insertion rate was 5.9% and the deletion rate was 4.2%. In the case of context-dependent models the rates were 23.4%, 5.7% and 3.8%, respectively.

As the table shows, our recognizer had the greatest difficulty with the novel *White Fang* (by Jack London). We suspect that this is partly due to the efforts of the speaker to render the dialects of the different characters and to the fact that the language model is thrown off by the spelling conventions which the author uses to the same effect. (For instance, some characters say "mebbe" rather than "maybe".) In any case, a language model trained on newspaper data is bound to have difficulties with a novel whose hero is a wolf. The perplexity [2] of the *White Fang* data, as calculated with the newspaper language model, turns out to be 1,743 which is extremely high. (For comparison the perplexity of *Washington Square* is 576 and the average perplexity of the test data used in our isolated word recognition experiments is 670.)

The results in Table 1 are the fruit of a long series of experiments. In the case of *White Fang*, the recognition rate on our very first experiment using context-independent models was 40%; screening the training data and smoothing the estimate of the covariance matrix of the silence model gave 52%; treating compound proper names such as *White Fang*, *Lip Lip* and *Gray Beaver* as single words rather than sequences of two words for the purpose of calculating language model scores increased the accuracy to 54%; some alignment errors made by the training algorithm were corrected by detecting long silences in a pre-processor, giving an accuracy of 58%; finally, context-dependent models increased the accuracy to 68%.

An experiment which did *not* lead to improvements in recognition accuracy concerned the imposition of minimum duration constraints on phoneme durations. We found that the use of such constraints was very effective in isolated word recognition (Gupta et al., 1992b) and one of

[2] The test set perplexities P were calculated in the usual way, using the formula $P = \Pr(w_1 \cdots w_N)^{-1/N}$, where $w_1 \cdots w_N$ is the string of words in the test set. This way of calculating perplexities suffers from the disadvantage that the scoring mechanism for words not included in the training data for the language model is somewhat arbitrary. However, this fact does not seem to account for the very high perplexity of *White Fang* compared to *Washington Square* since the percentage of out-of-vocabulary words was 1.4% in the case of *White Fang* and 2.2% in the case of *Washington Square*.

the principal reasons for using point scores in our search strategy was to be able to impose such constraints in a computationally efficient way. However, the only advantage that we were able to get from using duration constraints in the continuous speech case was a slight reduction in the amount of computation.

For the experiments reported in Table 1, the recognizer was configured as follows. In order to keep the size of the stack within reasonable bounds, we had to use a block advance of only 10 frames (1 frame = 10 ms). The parameter Δ was set at 140 frames (but it turns out that we could have taken Δ to be 50 frames without incurring any penalty in accuracy). The maximum number of entries in the stack was set to 60 000. (Whenever this figure was attained, the size of the stack was cut back to 30 000.) The number of theories passed from one block to the next was 3000. Using context-independent models, the CPU time required to run the experiments on a HP 720 workstation was 120 times real time. With context-dependent models the CPU time needed increased to about 300 times real time. This increase is mostly due to the increased computation needed to evaluate the β^*'s. (But note that as mentioned above, this could have been substantially reduced by changing the parameter Δ which determines the overlap between successive blocks.)

5. Future work

We have reported the results of some pilot experiments on continuous speech recognition using a 60 000-word vocabulary and a trigram language model. Obviously, these results are only significant insofar as they indicate the types of improvement that will be needed if speech recognition on this scale is to become a practical reality. Our results show that, contrary to the case of isolated word recognition, we cannot hope using current modelling methods to obtain high recognition accuracies with context-independent phoneme models and a universal, domain-independent language model.

It is clear that we will have to greatly increase our allophone inventory. Given the size of the vocabulary, the robustness problem is obviously a major issue. (The number of triphones in the dictionary is about 17 000; this number increases by a factor of about two when triphones spanning word boundaries are included.) We are experimenting with an approach to generalized triphone modelling using multiple templates to score every possible triphone. These templates are context-dependent models where context is described specifying the values of phonetic features on the right and the left in varying degrees of precision. Included among the templates are context-independent models and triphone models (in the case where the triphone is observed in the training data) as well as a large number of intermediate models.

In (Kenny et al., 1992) we described how an admissible heuristic for searching with allophone models could be calculated without changing the topology of the graph G^*. However, we have decided not to pursue this since the evidence in favour of a two-pass approach to large-scale speech recognition problems now seems to be quite compelling (Schwartz et al., 1992). Accordingly, we are redesigning the search strategy so that in the first pass the data in a block is searched using a coarse set of allophone models to match surface form phonemic transcriptions and a coarse language model. This search (which is essentially the same as the algorithm reported here) generates a list of partial recognition hypotheses for the data in the block, each hypothesis consisting of a short word string (possibly empty) together with a partial transcription of a word which extends beyond the end of the block. These hypotheses are rescored in the second pass using fine allophone models, a fine language model and a phonological component which checks the consistency of the surface form transcriptions hypothesized by the first pass. (In the new algorithm, the first pass treats surface form transcriptions of the words in each hypothesis as occurring in free variation; the second pass rejects any hypothesis whose surface form transcription cannot be obtained by the application of

phonological rules to the base form transcriptions of the words in the hypothesis.)

We have to resolve the question of whether a fast match ought to be incorporated into the search. The role of a fast match is to conduct a crude search of the entire lexical tree every time a word boundary is hypothesized; the algorithm we have presented can be viewed as maintaining several copies of the lexical tree in memory and abandoning the search of a given copy of the lexical tree whenever the heuristic indicates that it would be more promising to search another. This approach was motivated by the need to avoid the computational overhead of running a fast match to completion every time it is invoked, but the memory requirements are so great that we were constrained to work with a very small block advance. This problem will be alleviated by using a coarse language model in the first pass, but we are also experimenting with a very effective and very fast approximate acoustic match which uses information extracted from searching a graph such as G^* to score partial transcriptions at a cost of a single floating point operation per phoneme (Kenny et al., 1993). In this approach all of the time alignment for the search is carried out in a pre-processing step. This enables us to use powerful graph search techniques (Nilsson, 1982; Kenny et al., 1994) that have not previously been applied in speech recognition.

Since we are incorporating a phonological component to convert base forms to surface forms and then scoring surface forms using context-dependent allophone models, the search problem becomes quite complex even in the training phase (especially since we want to be able to handle unsegmented training files of arbitrary length). To deal with this, we are designing the recognition search strategy so that it can accommodate an abstract language model; this enables us to perform the Viterbi alignment of the data in a training file by supplying a language model constructed from the words in the training script.

Finally, it is obvious that language model adaptation will make the recognition task we have set ourselves much easier. Preliminary results in this direction are reported in (Zhao et al., 1993).

6. References

S. Austin,. P. Peterson, P. Placeway, R. Schwartz and J. Vandergrift (1990), "Toward a real-time spoken language system using commercial hardware", *Proc. DARPA Speech and Natural Language Workshop*, June 1990.

S. Austin, R. Schwartz and P. Placeway (1991) "The forward–backward search algorithm", *Proc. Internat. Conf. Acoust. Speech Signal Process. 91*, pp. 697–700.

A. Averbuch et al. (1987), "Experiments with the Tangora 20000 word speech recognizer", *Proc. Internat. Conf. Acoust. Speech Signal Process. 87*, pp. 701–704.

L.R. Bahl, R. Bakis, P.V. de Souza and R.L. Mercer (1988), "Obtaining candidate words by polling in a large vocabulary speech recognition system", *Proc. Internat. Conf. Acoust. Speech Signal Process. 88*, pp. 489–492.

L.R. Bahl et al. (1989), "Large vocabulary natural language continuous speech recognition", *Proc. Internat. Conf. Acoust. Speech Signal Process. 89*, pp. 465–467.

L. Bahl, P. de Souza, P. Gopalakrishnan, D. Nahamoo and M. Picheny (1991), "Decision trees for phonological rules in continous speech", *Proc. Internat. Conf. Acoust. Speech Signal Process. 91*, pp. 185–188.

L.R. Bahl, S.V. De Gennaro, P.S. Gopalakrishnan and R.L. Mercer (1993), "A fast approximate acoustic match for large vocabulary recognition", *IEEE Trans. Speech and Audio Processing*, Vol. 1, No. 1, January 1993, pp. 59–67.

G. Boulianne, P. Kenny, M. Lennig, D. O'Shaughnessy and P. Mermelstein (1994), "Books on tape as training data for continuous speech recognition", *Speech Communication*, Vol. 14, No. 1, February 1994, pp. 61–70.

L. Deng, P. Kenny, M. Lennig, V. Gupta, F. Seitz and P. Mermelstein (1991), "Phonemic hidden Markov models with continuous mixture output densities for large vocabulary word recognition", *IEEE Trans. Signal Process.*, Vol. 39, pp. 655–658.

L. Fissore, P. Laface, G. Micca and R. Pieraccini (1989), "Lexical access to large vocabularies for speech recognition", *IEEE Trans. Acoust. Speech Signal Process.*, Vol. 37, No. 8, pp. 1197–1213.

V. Gupta, M. Lennig and P. Mermelstein (1988), "Fast search strategy in a large vocabulary word recognizer", *J. Acoust. Soc. Amer.*, Vol. 84, No. 6, pp. 2007–2017.

V. Gupta, M. Lennig and P. Mermelstein (1992a), "A language model for very large-vocabulary speech recognition", *Computer Speech and Language*, Vol. 6, pp. 331–344.

V.N. Gupta, M. Lennig, P. Mermelstein, P. Kenny, P.F. Seitz and D. O'Shaughnessy (1992b), "Use of minimum duration and energy contour for phonemes to improve large vocabulary isolated-word recognition", *Computer Speech and Language*, Vol. 6, pp. 345–359.

F. Jelinek (1969), "A fast sequential decoding algorithm using a stack", *IBM J. Research and Development*, Vol. 13, pp. 675–685.

F. Jelinek (1976), "Continuous speech recognition by statistical methods", *Proc. IEEE*, Vol. 64, pp. 532–556.

P. Kenny, S. Parthasarathy, V. Gupta, M. Lennig, P. Mermel-

stein and D. O'Shaughnessy (1991), "Energy, duration and markov models", *Proc. Eurospeech 91*, pp. 655–658.

P. Kenny, R. Hollan, G. Boulianne, H. Garudadri, M. Lennig and D. O'Shaughnessy (1992), "An A* algorithm for very large vocabulary continuous speech recognition", *Proc. DARPA Speech and Natural Language Workshop*.

P. Kenny, R. Hollan, V. Gupta, M. Lennig, P. Mermelstein and D. O'Shaughnessy (1993a), "A* – Admissible heuristics for rapid lexical access", *IEEE Trans. Speech and Audio Processing*, Vol. 1, No. 1, pp. 49–58.

P. Kenny, P. Labute, Z. Li, R. Hollan, M. Lennig and D. O'Shaughnessy (1993b), "A very fast method for scoring phonetic transcriptions", *Eurospeech 93*, September 1993, presented.

P. Kenny, P. Labute, Z. Li and D. O'Shaughnessy (1994), "New graph search techniques for speech recognition", *Internat. Conf. Acoust. Speech Signal Process. 94*, submitted.

F. Kubala and R. Schwartz (1990), "A new paradigm for speaker-independent training and speaker adaptation", *Proc. DARPA Speech and Natural Language Workshop*, June 1990.

M. Lennig, V. Gupta, P. Kenny, P. Mermelstein and D. O'Shaughnessy (1990), "An 86,000-word recognizer based on phonemic models", *Proc. DARPA Speech and Natural Language Workshop*, pp. 391–396.

M. Lennig, D. Sharp, P. Kenny, V. Gupta and K. Precoda (1992), "Flexible vocabulary speech recognition", *Proc. ICSLP 92*, October 1992, pp. 93–96.

H. Ney, R. Haeb-Umbach, B.-H. Tran and M. Oerder (1992), "Improvements in beam search for 10000-word continuous speech recognition", *Proc. Internat. Conf. Acoust. Speech Signal Process. 92*, pp. I-9–I-12.

N. Nilsson (1982), *Principles of Artificial Intelligence* (Tioga Publishing Company).

A. Paesler and H. Ney (1989), "Continuous-speech recognition using a stochastic language model", *Proc. Internat. Conf. Acoust. Speech Signal Process. 89*, May 1989.

D. Paul (1991), "Algorithms for an optimal A* search and linearizing the search in the stack decoder", *Proc. Internat. Conf. Acoust. Speech Signal Process 91*, pp. 693–696.

S. Sagayama (1991), "A matrix representation of HMM-based speech recognition algorithms", *Proc. Eurospeech 91*, pp. 1225–1228.

R. Schwartz, A. Austin, F. Kubala, J. Makhoul, L. Nguyen, P. Placeway and G. Zavaglios (1992), "New uses for the *N* best sentence hypotheses within the Byblos speech recognition system", *Proc. Internat. Conf. Acoust. Speech Signal Process. 92*, pp. I-1–I-4.

F. Seitz, V. Gupta, M. Lennig, P. Kenny, L. Deng, D. O'Shaughnessy and P. Mermelstein (1990), "A dictionary for a very large vocabulary word recognition system", *Computer Speech and Language*, Vol. 4, pp. 193–202.

F.K. Soong and E.-F. Huang (1991), "A tree-trellis based fast search for finding the *N* best sentence hypotheses in continuous speech recognition", *Proc. Internat. Conf. Acoust. Speech Signal Process. 91*, pp. 705–708.

R. Zhao, P. Kenny, P. Labute and D. O'Shaughnessy (1993), "Issues in large scale statistical language modelling", *Eurospeech 93*, September 1993, presented.

V. Zue et al. (1991), "Integration of speech recognition and natural language processing in the MIT VOYAGER system", *Proc. Internat. Conf. Acoust. Speech Signal Process. 91*, pp. 713–716.