# A FAST LATTICE-BASED APPROACH TO VOCABULARY INDEPENDENT WORDSPOTTING*

*D.A. James & S.J. Young*

Cambridge University Engineering Department
Trumpington Street
Cambridge
CB2 1PZ
United Kingdom

## ABSTRACT

Practical applications of wordspotting, such as spoken message retrieval and browsing, require the ability to process large amounts of speech data at speeds many times faster than real-time. This paper presents a novel approach to this problem in which all of the stored audio material is preprocessed off-line to generate a phoneme lattice. At search time, putative word matches are found in this lattice using symmetric dynamic programming. The paper presents the details of the algorithms used and compares performance with a number of conventional approaches using a 20 keyword vocabulary on the DARPA Resource Management Task. The results show that the proposed method is very much faster yet performs acceptably compared to conventional systems which depend on keyword-specific training or prior knowledge of the test set vocabulary.

## 1. INTRODUCTION

In recent years, computers have become increasingly able to manipulate non-textual data, and applications such as video and voice mail have arisen to take advantage of this new processing capability. A recent experimental system has been *Pandora*, developed by Olivetti Research Ltd in Cambridge, UK [1]. In the Pandora system, Unix workstations, augmented by cameras and microphones, are connected by a wide-bandwidth optical fibre-based network, providing facilities for video telephony and messaging. However, in practice, the utility of Pandora has been compromised by the lack of methods for browsing large amounts of stored video data for items of particular interest to the user. Clearly, an accurate system for automatically indexing this data and retrieving specific items would greatly benefit users.

There have been several recent attempts to solve the problem of how to retrieve stored speech data [2] [3] [4]. Of these, the most novel approach is that presented in [2], in which speech recognition is performed on subword speech models, called "features", extracted from a task-dependent vocabulary obtained from radio news broadcasts. The authors claim that a relatively small domain-dependent feature set (around 800 features) is sufficient to index even a large collection of messages. The process of message retrieval then reduces to a match of the feature decomposition of a set of keywords against the recogniser transcriptions. This method allows for instantaneous matching of keywords against the unknown speech. In simulation, good retrieval performance is achieved, but in practice, it is difficult to achieve good recognition accuracy on such short acoustic units. A retrieval system based on conventional wordspotting, using more robust recognition units, would produce better results.

In message retrieval generally, a major problem is the tradeoff between speed of retrieval and flexibility of keyword choice. Assuming a fixed vocabulary and running a standard continuous speech recogniser once over all the unknown speech, wordspotting reduces to searching the output transcription files. This approach has been employed in recently reported wordspotters which treat the problem as a special case of large vocabulary continuous speech recognition [5]. The disadvantage of this approach is that the vocabulary is fixed in advance, and any word not appearing in this vocabulary cannot be used to search through the message database. At the other extreme, total flexibility of keyword choice necessitates re-running the wordspotter every time a new keyword is specified. This results in unacceptably slow performance, especially for a large database of messages.

Clearly, it is desirable to formulate an approach to the problem which combines flexibility of keyword selection with fast searching. A possible solution to the problem would be to obtain a vocabulary independent transcription of the spoken content of a message. This could be computed and stored in a compact form once and for all for each message, and searched quickly at any time for any keyword or keywords specified by the user. There would thus be no need to rerun a slow keyword-dependent wordspotter every time the user wished to search on new keywords.

In this paper we describe such a wordspotter. We address the requirements of keyword flexibility and fast performance by using a modified Viterbi HMM-based phone recogniser to obtain a compact intermediate decomposition of an utterance. This intermediate form is a phone lattice, in which multiple phone hypotheses are stored for every point throughout the speech. The wordspotting stage then becomes a symmetric dynamic programming match of the keyword pronunciation against each lattice, with penalties for phone insertion, deletion or substitution. Subsequent sections of this paper deal with the generation of phone lattices, the keyword matching criteria, and a comparison of lattice-based wordspotter results with those obtained by

*TO APPEAR, PROC. ICASSP 1994, ADELAIDE.

conventional approaches on a 20 word subset of the DARPA Naval Resource Management Task.

## 2. OVERVIEW OF LATTICE GENERATION AND KEYWORD MATCHING

A lattice is defined formally as a connected loop-free directed graph. If we associate each node of the graph with a point of time during the utterance, and label each edge with a phone hypothesis and a score representing the likelihood of that hypothesis, then the lattice can be used for the storage of multiple potential model sequences output by a continuous speech recogniser. Here, phone lattices are generated from a modified Viterbi speech recogniser based on the Token Passing Paradigm [6]. The recognition network simply consists of all phone models arranged in parallel. At every speech frame, phone hypotheses are updated and the identities of the top $N$ phones ending at that frame are stored, where $N$ is the *degree*, or "depth", of the required lattice. The best phone hypothesis is propagated around to the start of the network. When all the speech data has been used up, the resulting lattice is pruned so that at most $N$ edges begin at any lattice node. Clearly, a lattice of degree 1 is the same as the maximum likelihood phone sequence generated for an utterance. Figure 1 illustrates a section of a lattice around an occurrence of the word "ship". Wordspotting can now be performed on

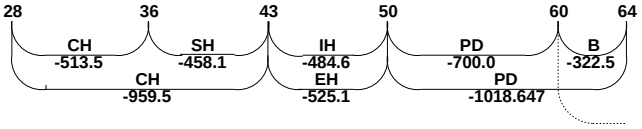


Figure 1. Section of lattice of degree 2 containing keyword *ship*

the utterance by a DP match of the keyword pronunciation against the lattice. Phones may be inserted, deleted or substituted to obtain a valid path for the keyword through the lattice. Formally, we define a lattice edge,

$$l \quad = \quad (p_b, p_e, p, s)$$

where $p$ is a recognised phone, $p_b$ and $p_e$ are the indices of its beginning and end frames and $s$ is the score output by the recogniser. If $L$ is the set of lattice edges, and we define a function

$$V(b,e,p) = \begin{cases} s & \text{if } (b,e,p,s) \in L \\ -\infty & \text{otherwise} \end{cases}$$

then the cumulative DP matching function $C(i,e)$, which returns the best path score for keyword phones $p_1 \ldots p_i$ between some possible keyword start time $t$ and time $e$, can be defined as follows:

$\forall t, C(0,t) = 0$, and

$$C(i,e) = \max_b \begin{cases} C(i-1,b) + V(b,e,p_i), \\ C(i-1,b) + (1+e-b)P_s \\ \quad + \max_z V(b,e,z), \\ C(i,b) + P_i, \\ C(i-1,e) + P_d \end{cases}$$

where $P_i, P_d$ and $P_s$ are empirically chosen penalties for phone insertion, deletion and substitution respectively.

In practice, it these operations are constrained to cut down on the amount of computation time and reduce the number of false alarms generated. For example, in the lattice search implemented here all keyword phones are labelled either as "weak" or "strong". Strong phones are those which must be matched exactly against a lattice edge for the construction of the keyword path to continue, whereas weak phones may be deleted or substituted. Phone substitutions are further constrained by the division of the phone set into 7 broad acoustic-phonetic classes, and a rule stating that a phone may only be substituted for another member of its broad class. A phone can only be inserted into a keyword path if it is of minimal duration, subsuming only two speech frames, and at most one such phone may be inserted between two keyword phones. If a successful path is constructed for a keyword, it is assigned a score which is the ratio of the keyword path score $C(1,n)$ to the maximum likelihood score for the unknown speech over the same time interval. Thus the ratio score can be thought of as a measure of the "depth" of the keyword path in the two-dimensional lattice representation of speech.

## 3. EXPERIMENTS

Experiments were carried out on the DARPA Resource Management (RM) task. The CMU pronunciation dictionary and phone set were used [7]. Sets of monophone and triphone models, created using the standard RM SI-109 training set, were used. All HMM training used *HTK*, the portable HMM toolkit developed within the CUED Speech Group. Wordspotting was carried out on the speaker independent February and October 1989 test sets, representing an overall test set of 34 minutes of speech. The keyword set was chosen to have roughly the same distribution of keyword lengths and potential keyword confusions as in the standard Road Rally task. The test set contained a total of 494 keyword occurrences. The keyword set is illustrated in Table 1. Several different wordspotting experiments were

| | | | |
|---|---|---|---|
| asw | equipment | propulsion | seventeen |
| capacity | fuel | rating | ship |
| casualty | maximum | readiness | sub |
| category | mission | report | submarine |
| display | port | sassafras | twenty |

Table 1. Twenty word keyword set for RM experiments

performed. Results from the lattice-based wordspotter are compared with more conventional approaches ranging from a whole-word model wordspotter to a phone-based connected word recogniser. Viterbi experiments where no vocabulary knowledge in used to constrain the garbage model are labelled as *VOCIND*; those where explicit vocabulary knowledge is included are labelled *VOCDEP*. For VOCIND experiments in which the keyword models were concatenated from monophones, it was first necessary to devise a method for creating garbage models from the set of monophones.

### 3.1. Monophone Clustering for Garbage Model Creation

Conventional wordspotters generally use a number of garbage models trained from non-keyword speech. However, since there is no distinction here between keyword and non-keyword speech in the training set, a novel method of creating garbage models was required. The method used in these experiments derives garbage models from the monophones by pooling all monophone states and selecting some number $n$ of these states using an agglomerative clustering procedure. The clustering algorithm reduces the number of initial states by iteratively merging pairs of states until the number of states has reached the required value of $n$. With each iteration a pair of states $i, j$ is chosen to maximise a similarity measure defined as follows:

$$s_{i,j} = \sum_{m=1}^{M} \log b_j(\mu_{im}) + \log b_i(\mu_{jm})$$

where $M$ is the number of Gaussian mixtures used to model the output probability distribution of each HMM, $\mu_{im}$ is the mean vector of the $m$th mixture of each model i and $b_j(x)$ is the probability of observation vector $x$ being produced by state $j$. A many-to-one mapping is thus gradually constructed from the set of monophone states to the set of $n$ clustered states, and the set of garbage models is created by replacing each monophone state by its image under this mapping. Therefore for every monophone there is a corresponding garbage model and these are placed in parallel to create the complete garbage model. Varying $n$ provides a way of controlling the operating point of the system; increasing $n$ makes the garbage model closer to the set of monophones and thus lowers the number of keyword hypotheses produced.

### 3.2. Multiple Mixture Monophone System

48 3-state single mixture monophone models were initialised from existing TIMIT HMMs. These models were iteratively re-estimated on the RM training set and had their number of Gaussian mixture distributions increased up to 8. These models were then used in wordspotting to test the performance of the lattice-based approach compared to conventional systems. In an initial VOCIND experiment, a non-keyword model was created using the procedure described above, with the initial set of 144 monophone states being clustered to produce 80 states. The clustered state monophones, the image of the original monophone set under the clustering, were placed in parallel with the keywords to produce the recognition network. In the next experiment, a level of vocabulary knowledge was introduced by using the original set of 48 monophones to create word models for the most common 80 words observed in the test data. Finally the lattice method was used, with all lattices being generated with degree 8. To allow for fair comparison between network-based and lattice-based wordspotters, the network-based results were improved by rescoring all putative keywords to obtain a ratio score, as in [8]. The results are obtained as a standard NIST Figure of Merit averaged across 0 to 10 false alarms per keyword per hour and are shown in Table 1. It can be seen that the lattice wordspotter performs reasonably well, its FOM being roughly the same as that of a system which incorporated particular knowledge of the test set vocabulary.

| Wordspotter | FOM |
|---|---|
| Network – Clustered VOCIND garb model | 58.85 |
| Network – 80 word VOCDEP garb model | 66.93 |
| Lattice – Symmetric DP search | 66.95 |

Table 2. FOM Results for monophone-based experiments

### 3.3. Triphone Based System

Further experiments were performed using two sets of more sophisticated phone models. These were whole word models, and 6-mixture state-clustered triphones with function word dependent phone models[1]. A full account of the triphone clustering and re-estimation process can be found in [9]. The goal of these experiments was to compare the performance of the lattice system with wordspotters which take specific account of knowledge about the keyword set or the overall task vocabulary. Keyword dependence was modelled by using the whole word models, which were obtained by concatenating appropriate 4-mixture monophone models and re-estimating solely on whole word examples of the keywords from the training data. In VOCDEP experiments, vocabulary dependence was implemented by building a "no-grammar" network consisting of the 971 non–keywords in the RM lexicon, concatenated from the appropriate sub–word models. In VOCIND experiments, the sub–word models are simply placed in parallel to create the garbage model. Results, in terms of the wordspotter FOM, were obtained for each of these experiments and for the triphone-based lattice wordspotter and are given in Table 2. It can be seen

| Keywords | Garbage Model | FOM |
|---|---|---|
| Monophone | VOCIND – Clustered mono. | 58.85 |
| | VOCDEP – Monophone | 64.22 |
| Triphone | VOCIND – Monophone | 73.79 |
| | VOCDEP – Triphone | 89.02 |
| Whole Word | VOCIND – Monophone | 76.09 |
| | VOCDEP – Triphone | 83.53 |
| Triphone-based Lattice Wordspotter | | 76.75 |

Table 3. FOM Results for network and lattice wordspotters

from the table that the triphone-based lattice wordspotter achieves a respectable level of performance, given its independence of the test set vocabulary and the keyword set, marginally outperforming the Whole Word/VOCIND system. It is interesting that the FOM for the VOCDEP system decreases when whole word keyword models are used instead of triphone-based word models. This is because the improvement obtained for the VOCIND experiments using whole word models is due to the significant reduction obtained in the number of false alarms. However, when the non-keyword vocabulary is explicitly modelled, there is a much smaller number of false alarms to begin with, and the use of whole word models, some of which are poorly trained, leads to an overall decrease in performance. A solution to this problem would be to use triphone-based word

[1] Although the function word dependent models were not used in lattice creation.

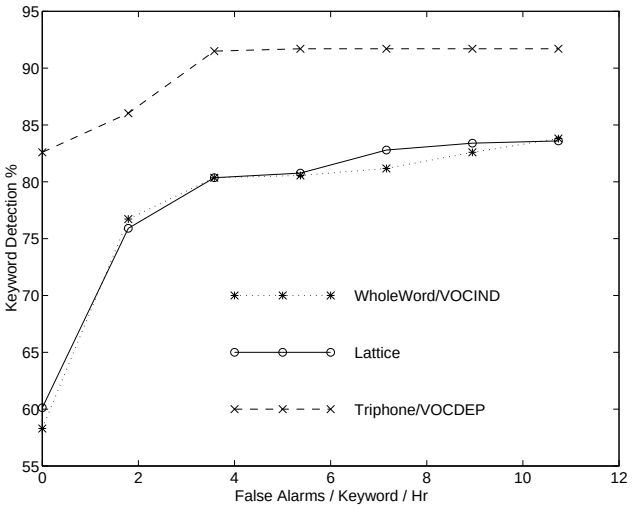


Figure 2. Tradeoffs for triphone and whole word experiments

models for those keywords with a small number of training tokens in the training set, and whole word models for all other keywords. Figure 2 shows the tradeoff curves obtained in these experiments. It is also interesting to note that the monophone/VOCDEP results are slightly inferior to those reported earlier for the 80 word garbage model experiment. This would seem to support the recent observation that varying non-keyword vocabulary size does not improve wordspotter performance a great deal [5]. Recently reported improvements in wordspotter performance based on the use of a language model are probably due more to higher-level grammatical constraints (such as the use of a bigram or word-pair grammar to constrain word sequences) rather than better models of the task vocabulary.

Owing to the relatively poor accuracy of the monophones used to generate the initial set of lattices, it was found necessary to hand-tune the strong/weak labelling of phones in the keyword pronunciations to achieve good performance. This was not necessary in experiments on the triphone-based lattices as phone accuracy was far better, and labellings were generated simply by marking all phones in all stressed syllables, and the first phone in any unstressed syllable, as strong. Thus such labellings could be generated rapidly and automatically from a large on-line phonetic dictionary of English.

### 3.4. Speed of Execution

The primary advantage of the lattice-based search approach is the execution speed once the lattice has initially been compiled and stored. The lattice-based search for 20 keywords runs 60 times faster than real-time on a Silicon Graphics Indigo workstation. If in triphone-based experiments, keyword pronunciations are constrained by marking all phones as strong, the search speed rises to 95 times real-time with only a 2% drop in the Figure of Merit. A search on one keyword of average length runs at 360 times real-time. A wordspotter such as this has applications in the problem of indexing and retrieval of stored voice or video mail, a task in which no assumptions can be made about the overall vocabulary and the keyword vocabulary changes with every new query.

## 4. CONCLUSION

In this summary we have presented a novel approach to the problem of faster than real-time searching through a corpus of speech for a set of arbitrary keywords. The lattice-based method represents an efficient solution, with the speech recognition component of the search performed only once for every utterance, and respectable wordspotter performance obtainable thereafter much faster than real-time. We are confident that several computational savings can be made in our wordspotter, and are currently using the lattice-based approach in retrieval experiments on a corpus of news stories obtained from radio broadcasts.

## ACKNOWLEDGEMENT

D. A. James is funded by a CASE Research Studentship, and gratefully acknowledges the assistance and support of the Science and Engineering Research Council and Olivetti Research Limited.

## REFERENCES

[1] A. Hopper, *Pandora – An Experimental System for Multimedia Applications*, Operating Systems Review, Vol. 24,No. 2, April 1990.

[2] U. Glavitsch, P. Schäuble, *A System for Retrieving Speech Documents*, Proc. SIGIR 1992, pp168-176.

[3] R.C. Rose, E.I. Chang, R.P. Lippmann, *Techniques for Information Retrieval from Voice Messages*, Proc. Int. Conf. Acoust., Speech and Sig. Processing, 1991, pp317-320.

[4] J.R. Rohlicek, D. Ayuso, M. Bates, R. Bobrow, A. Boulanger, H. Gish, P. Jeanrenaud, M. Meteer, M. Siu, *Gisting Conversational Speech*, Proc. Int. Conf. Acoust., Speech and Sig. Processing, 1992, pp II-113-117.

[5] M. Weintraub, *Keyword-Spotting Using SRI's Decipher Large Vocabulary Speech Recognition System*, Proc. Int. Conf. Acoust., Speech and Sig. Processing, 1993, pp II-463-466.

[6] S.J. Young, N.H. Russell, J.H.S. Thornton, *Token Passing: A Simple Conceptual Model for Connected Speech Recognition Systems*, Cambridge University Engineering Department Technical Report F/INFENG/TR.38, July 1989.

[7] K.-F. Lee, Automatic Speech Recognition: The Development of the SPHINX System, Kluwer Academic Publishers, Boston.

[8] R.C. Rose, D.B. Paul, *A Hidden Markov Model Based Keyword Recognition System*, Proc. Int. Conf. Acoust., Speech and Sig. Processing, May 1990, pp129-132.

[9] S.J. Young, P.C. Woodland, *The Use of State Tying in Continuous Speech Recognition*, Proc. Eurospeech, Berlin 1993.

# A FAST LATTICE-BASED APPROACH TO VOCABULARY INDEPENDENT WORDSPOTTING[2]

*D.A. James & S.J. Young*
Cambridge University Engineering Department
Trumpington Street
Cambridge
CB2 1PZ
United Kingdom

Practical applications of wordspotting, such as spoken message retrieval and browsing, require the ability to process large amounts of speech data at speeds many times faster than real-time. This paper presents a novel approach to this problem in which all of the stored audio material is preprocessed off-line to generate a phoneme lattice. At search time, putative word matches are found in this lattice using symmetric dynamic programming. The paper presents the details of the algorithms used and compares performance with a number of conventional approaches using a 20 keyword vocabulary on the DARPA Resource Management Task. The results show that the proposed method is very much faster yet performs acceptably compared to conventional systems which depend on keyword-specific training or prior knowledge of the test set vocabulary.