DYNAMIC MATCH PHONE-LATTICE SEARCHES FOR VERY FAST AND ACCURATE UNRESTRICTED VOCABULARY KEYWORD SPOTTING

K. Thambiratnam and S. Sridharan

Speech and Audio Research Laboratory
Queensland University of Technology
GPO Box 2434, Brisbane, Australia 4001
[k.thambiratnam,s.sridharan]@qut.edu.au

ABSTRACT

The ability to search for typically out-of-vocabulary terms such as names, acronyms and foreign words is a requirement of many audio indexing applications. To date, such applications have employed unrestricted vocabulary keyword spotting approaches that unfortunately suffer from poor miss rates or slow query speeds. This paper proposes a very fast and accurate keyword spotting approach named Dynamic Match Phone-Lattice keyword Spotting. Reported experiments on conversational telephone speech and microphone speech demonstrate that the proposed method dramatically outperforms conventional methods and is capable of searching at speeds in excess of 300 times real-time while maintaining low miss rate performance.

1. INTRODUCTION

The ever-increasing volume and importance of audio and multimedia data has brought with it the need for rapid audio indexing technologies. Fast and accurate Large Vocabulary Continuous Speech Recognisers (LVCSR) have provided an intermediary solution by transcribing speech to text that can then be rapidly searched using conventional text search engines. However such systems are severely restricted by the vocabulary of the LVCSR engine.

Many applications, such as surveillance and news-story indexing, require support for typically out-of-vocabulary keyword queries such as names, acronyms and foreign words. In such cases, unrestricted vocabulary keyword spotting methods such as HMM-based keyword Spotting (HMMS) [1] have provided a solution, though at the expense of considerably slower query speeds. Faster approaches such as Phone-Lattice keyword Spotting (PLS) [2] offer significantly quicker spotting but are encumbered by poor miss rate performance.

This paper proposes a very fast and accurate keyword spotting method named Dynamic Match Phone-Lattice keyword Spotting (DMPLS). DMPLS builds upon lattice-based methods, such as PLS, but addresses the issue of inherent phone recogniser errors that adversely affect miss rate performance. This is done by augmenting the lattice search with dynamic programming (DP) sequence matching techniques to provide robustness against erroneous phone lattice realisations.

Subsequent sections of this paper discuss the proposed DMPLS algorithm, the conventional HMM-based and the lattice-based methods, as well as the results of keyword spotting experiments on conversational telephone speech and microphone speech.

2. BACKGROUND

This section briefly describes the HMMS and PLS methods used as baseline systems for the experiments reported in this paper.

2.1. HMM-based Keyword Spotting

HMM-based Keyword Spotting uses a HMM-based speech recogniser to postulate candidate occurrences of a target keyword in continuous speech. All non-target keywords in the target domain's vocabulary are represented by a single 'non-keyword' word. An open word-loop recognition network containing the target keywords and the non-keyword word is then constructed and used in recognition to generate a time-marked sequence of keyword and non-keyword tokens for a given observation sequence. Optionally thresholding on the duration-normalised output likelihood of word tokens is applied to reduce false alarm (FA) rates.

Although a plethora of non-keyword models have been proposed in literature, the speech background model described in [3] is used as the non-keyword model in this paper due to its prevalent use in many other areas of speech research.

2.2. Phone-Lattice Keyword Spotting

The Phone-Lattice keyword Spotting approach, PLS, proposed in [2] achieves significantly faster query speeds than HMMS by first indexing speech files for subsequent rapid searching. For each speech file, a phone-lattice representation of the speech is generated by performing a N-best Viterbi recognition pass. The resulting phone-lattices compactly encode multiple observed phone sequence hypotheses for any particular region in the speech.

At query time, keyword spotting then only requires searching of the previously prepared phone-lattices. For each node in a lattice, the lattice is traversed backwards to obtain a list of all phone sequences that terminate at the node. Any such phone sequences that match the target phone sequence are emitted as hypothesised keyword occurrences. Since this entire lattice traversal process operates on text, the operation is very fast. Optimisations to improve the speed of PLS are described in [2].

3. DYNAMIC MATCH PHONE-LATTICE KEYWORD SPOTTING

Dynamic Match Phone-Lattice keyword Spotting is an extension of PLS that uses the Minimum Edit Distance [4] (MED) during lattice searching to compensate for phone recogniser insertion, dele-

0-7803-8874-7/05/$20.00 ©2005 IEEE

tion and substitution errors. This addresses a shortcoming of PLS — the requirement for the target phone sequence to appear in its entirety within the phone-lattice for consideration as a hypothesised keyword occurrence. Such a requirement poses significant performance implications since PLS is based on phone recognisers that inherently suffer from high insertion, deletion and substitution error rates. This was confirmed in preliminary investigations that found that target phone sequences were frequently erroneously realised within the phone-lattice, even when considering the multiple hypotheses encoded within the lattice.

Given source and target sequences, the MED calculates the minimum cost of transforming the source sequence to the target sequence using a combination of insertion, deletion, substitution and match operations, where each operation has an associated cost. In DMPLS, each observed lattice phone sequence is scored against the target phone sequence using the MED. Lattice sequences are then accepted/rejected by thresholding on the MED score, hence providing robustness against phone recogniser errors. PLS is a special case of DMPLS where a threshold of 0 is used.

The capability to remain robust against phone recogniser errors has the potential to yield improved keyword spotting performance. For example in preliminary experiments it was found that the phone sequence 'k ae p t ih n' (the word 'CAPTAIN') was often erroneously realised within the phone-lattices as the phone sequence 'k ae p ih t ih n'. Whereas the conventional PLS method would have excluded these observed sequences, DMPLS simply assigns them a non-zero sequence matching score (in this case a score equal to the cost of inserting the phone 'ih'). Similar cases were observed for phone recogniser substitution errors.

Another DP inspired PLS approach was previously proposed in [5]. This method differs from DMPLS as it applies a dynamic programming match on a path/utterance level. In contrast, DMPLS performs dynamic matching on a localised per-word scale. It is believed that the DMPLS approach is better suited to the spotting task as keyword spotters seek to discriminate keywords from non-keywords on a localised scale.

3.1. Basic Dynamic Match Phone-Lattice Keyword Spotting

Let $P = (p_1, ..., p_N)$ be defined as the target phone sequence, where N is the target phone sequence length. Additionally let S_{max} be the maximum MED score threshold, K be the maximum number of observed phone sequences to be emitted at each node, and V be defined as the number of tokens used during lattice traversal. Then for each node in the phone-lattice, where node list traversal is done in time-order:

1. For each token in the top K scoring tokens in the current node:

    (a) Let $Q = (q_1, ..., q_M), M = N + MAX(C_i) * S_{max}$ be the observed phone sequence obtained by traversing the token history backwards M levels, where C_i is the insertion cost function.

    (b) Let $S = MED(Q, P, C_i, C_d, C_s)$, where C_d is the deletion cost function, C_s is the substitution cost functions, and $MED(\ldots)$ returns the score of the first element in the last column of the MED cost matrix that is $\leq S_{max}$ (or ∞ otherwise).

    (c) Emit Q as a keyword occurrence if $S \leq S_{max}$

2. For each node linked to the current node, perform V-best token set merging of the current node's token set into the target node's token set.

3.2. Optimised Dynamic Match Phone-Lattice Search

A number of optimisations can be used to improve throughput of DMPLS. In particular, MED calculations can be aggressively optimised to reduce processing time. One such optimisation is to only calculate successive columns of the MED matrix if the minimum element of the current column is less than S_{max}, since by definition the minimum of a MED matrix column is always greater than or equal to the minimum of the previous column. This optimisation gave a 35% relative increase in throughput in preliminary experiments. Other MED optimisations can be applied though these remain beyond the scope of this paper. For experiments reported in this paper, no MED optimisations were used.

Another optimisation is the removal of lattice traversal from query-time processing. Since the paths traversed through the lattice are independent of the queried phone sequence (traversal is done purely by maximum likelihood), it is possible to perform the lattice traversal during the speech preparation stage and hence only store the observed phone sequences at each node for searching at query-time. Therefore, if the maximum query phone sequence length is fixed at N_{max} and the maximum sequence match score is preset at S_{max}, it is only necessary to store observed phone sequences of length $M_{max} = N_{max} + MAX(C_i) * S_{max}$ for searching at query time. DMPLS query-time processing then reduces to simply calculating the MED between each stored observed phone sequence and the target phone sequence.

4. EXPERIMENTAL PROCEDURE

Experiments were performed to compare the keyword spotting and time performance of DMPLS and HMMS on evaluation sets taken from the Switchboard-1 (SWB1) conversational telephone speech corpus and the TIMIT microphone speech database.

16-mixture triphone HMM acoustic models and a 256-mixture Gaussian Mixture Model background model were trained on a 150 hour subset of SWB1 speech for use with HMMS and DMPLS SWB1 evaluations. Similar models were also trained on the training subset of the Wall Street Journal 1 (WSJ1) microphone speech database for experiments on TIMIT. Additionally 2-gram and 4-gram phone-level language models were trained on SWB1 and WSJ1 for use during the lattice generation stage of DMPLS. All speech was parameterised using Perceptual Linear Prediction coefficient feature extraction and Cepstral Mean Subtraction.

4.1. DMPLS experimental setup

DMPLS experiments were performed using the optimisations detailed in section 3.2. During the preparation stage, lattices were generated for each utterance by performing a U-token Viterbi decoding pass and a 2-gram phone-level language model. The resulting lattices were then expanded using a 4-gram phone-level language model and pruned using a beam-width of W to reduce the complexity of the lattices. Finally a second V-token traversal was performed to generate the top 10 scoring observed phone sequences of length 11 at each node (allowing spotting of sequences of up to $(11 - MAX(C_i) * S_{max})$ phones).

MED calculations used a fixed deletion cost of $C_d = \infty$ as preliminary experiments found that poor results were obtained for non-infinite values of C_d. The insertion cost was also fixed at $C_i = 1$. However C_s was allowed to vary based on phone substitution rules. Although full specification of the substitution rules is beyond the scope of this paper, the basic rules used for determining C_s were: $C_s = 0$ for same-letter consonant phone substitution (eg. 'n' and 'nx', 'z' and 'zh'), $C_s = 1$ for vowel substitutions, $C_s = 1$ for closure and stop substitutions and $C_s = \infty$ for all other substitutions. These rules for C_s were motivated by trends observed in the confusion matrix of the phone recogniser used for lattice generation. The sequence matching threshold, S_{max}, was fixed at 2 for all experiments unless noted otherwise.

4.2. Evaluation procedure

A keyword evaluation set was created for each evaluation database. The choice of query words was constrained to words that had 6-phone-length pronunciations to reduce the scope of experiments. This was done because keyword spotting performance generally varies with target keyword length and it was not possible to use the same query word set across evaluation sets.

For the SWB1 evaluations, approximately 2 hours of speech was labeled as evaluation speech. From this speech, 360 6-phone-length unique words were randomly chosen and labeled as query words. These query words appeared a total of 808 times in the evaluation speech. In a similar fashion, 1 hour of speech was taken from the TIMIT test database (excluding SA1 and SA2 utterances) and labeled as TIMIT evaluation data. Then, 200 6-phone-length unique query words with a total of 480 occurrences in the evaluation data were labeled as TIMIT evaluation query words.

The systems were evaluated by performing single-word keyword spotting for each query word across all utterances in a given evaluation set. The total miss rate for all query words and the false alarm per keyword occurrence (FA/kw) rate were then calculated using reference transcriptions of the evaluation data. Additionally the total CPU processing minutes per queried keyword per hour (CPU/kw-hr) was measured for each experiment using a 3GHz Pentium 4 processor. For DMPLS, CPU/kw-hr only included the CPU time used during the DMPLS search stage. For all experiments a commercial-grade decoder was used to ensure that the best possible CPU/kw-hr results were reported for HMMS (HMMS time performance is bound by decoder performance).

5. RESULTS AND DISCUSSION

To aid discussion the notation DMPLS[U,V,W,S_{max}] is used to specify DMPLS configurations, where U is the number of tokens for lattice generation, V is the number of tokens for lattice traversal, W is the pruning beamwidth, and S_{max} is the sequence match score threshold. The notation HMMS[α] is used when referring to HMMS configurations where α is the duration-normalised output likelihood threshold used.

Although a high FA/kw rate is undesirable in keyword spotters, a subsequent keyword verification stage (eg. [6]) can always be used to cull extraneous FAs. As such, discussion focuses on trends in miss rate. However some consideration is given to very high FA rates as a large number of FAs translates to higher post-verification FA rates as well as a decrease in query execution time (due to greater keyword verifier processing burden).

5.1. Microphone speech experiments

Baseline performance for HMMS, PLS and DMPLS were first measured on the TIMIT evaluation set and are shown in Table 1. The DMPLS[3,10,200,2] configuration was arbitrarily chosen as the baseline DMPLS configuration. PLS results were obtained by using a special DMPLS[3,10,200,0] setup with all MED cost functions set to ∞, simulating the exact matching nature of PLS. PLS timings are not reported since a true system was not used.

Method	Miss Rate	FA/ kw	CPU/ kw-hr
HMMS[∞]	1.6	44.2	1.58
HMMS[-7580]	10.4	36.6	1.58
HMMS[-7000]	39.8	16.8	1.58
PLS[3,10,200,0]	32.9	0.4	-.–
DMPLS[3,10,200,2]	10.2	18.5	0.30

Table 1. Baseline keyword spotting results evaluated on TIMIT

The timing results demonstrated that as expected DMPLS was significantly faster than the baseline HMM-based HMMS method, running at approximately 5 times the speed. This amounts to a baseline DMPLS system capable of searching 1 hour of speech in 18 seconds, confirming the suitability of DMPLS for very fast keyword spotting. DMPLS also had more favourable FA/kw performance: at 10.2% miss rate, DMPLS had a FA/kw rate of 18.5, significantly lower than the 36.6 FA/kw rate achieved by the HMMS[-7580] system. However, HMMS was still capable of achieving a much lower miss rate of 1.6% using the HMMS[∞] configuration, though at the expense of considerably more FAs.

The miss rate achieved by the baseline lattice-based PLS system was very poor compared to that of DMPLS. This confirmed that the phone error robustness inherent in DMPLS yielded considerable performance benefits. Table 2 shows the results of experiments to quantify in isolation the contribution of the various DP aspects of the baseline DMPLS system; specifically the effects of insertions, same-letter consonant substitution, vowel confusions and stop/closure confusion.

Method	Miss Rate	FA/ kw
PLS[3,10,200,0]	32.9	0.4
DMPLS[3,10,200,2] insertions	28.5	1.2
DMPLS[3,10,200,2] same letter subst	31.0	0.5
DMPLS[3,10,200,2] vowel subst	15.6	7.8
DMPLS[3,10,200,2] closure/stop subst	23.5	3.0
DMPLS[3,10,200,2] baseline	10.2	18.5

Table 2. TIMIT performance when isolating various DP rules

The results show that insertion and same-letter consonant substitution rules only provided a small performance benefit over a PLS system, whereas vowel and closure/stop substitution rules yielded considerable gains in miss rate. FA/kw rates rose significantly when using vowel substitution, indicating that a more careful choice in vowel substitution rules may have provided lower FA/kw rates (eg. knowledge-based rules that only allowed substitution between highly confusable vowels).

Table 3 shows the results of experiments performed to measure the effects of various DMPLS parameters: number of lattice generation tokens, number of lattice traversal tokens, pruning beamwidth, and S_{max}. Although all parameters provided some tuning capability, S_{max}, pruning width and lattice generation tokens gave the best of control over FA/kw-hr, CPU/kw-hr and miss rate respectively. Performance was relatively insensitive to the number of lattice traversal tokens.

Parameter	Method	Miss Rate	FA/ kw	CPU/ kw-hr
Prune Width	DMPLS[3,10,150,2]	12.5	12.2	0.18
	DMPLS[3,10,200,2]	10.2	18.5	0.30
	DMPLS[3,10,250,2]	9.2	24.7	0.47
# Lat Gen Tokens	DMPLS[3,10,200,2]	10.2	18.5	0.30
	DMPLS[5,10,200,2]	5.8	38.4	0.71
# Lat Traverse Tokens	DMPLS[3,5,200,2]	10.4	17.4	0.28
	DMPLS[3,10,200,2]	10.2	18.5	0.30
	DMPLS[3,20,200,2]	9.8	18.8	0.29
S_{max} Threshold	DMPLS[3,10,200,0]	31.0	0.5	0.30
	DMPLS[3,10,200,1]	13.3	4.3	0.30
	DMPLS[3,10,200,2]	10.2	18.5	0.30
	DMPLS[3,10,200,3]	8.7	52.0	0.30

Table 3. Effect of DMPLS parameters, evaluated on TIMIT

Given the results of the tuning experiments, two tuned DMPLS systems were constructed. For both systems, the number of lattice generation tokens was increased to 5 to reduce miss rate. To compensate for the resulting speed decrease, the pruning width was reduced to 150. Finally S_{max} was adjusted for each system to tune the FA/kw rate. The performances of the tuned systems are shown in Table 4. DMPLS[5,10,150,2] provided a lower 7.3% miss rate (2.9% absolute gain) while keeping FA/kw and CPU/kw-hr rates close to the baseline DMPLS system. DMPLS[5,10,150,1] achieved a lower 5.6 FA/kw rate compared to the 18.5 baseline figure while maintaining comparable miss and CPU/kw-hr rates. The results demonstrate that DMPLS can be tuned for application specific requirements in a fairly intuitive fashion.

Method	Miss Rate	FA/ kw	CPU/ kw-hr
DMPLS[5,10,150,1]	11.5	5.6	0.31
DMPLS[5,10,150,2]	7.3	22.3	0.31

Table 4. Optimised DMPLS configurations, evaluated on TIMIT

5.2. Conversational telephone speech experiments

Results for HMMS, PLS, and DMPLS experiments on conversational speech SWB1 data are shown in Table 5. Of note is the dramatic increase in FA/kw rates for all systems compared to those observed for the TIMIT evaluations. This is most likely because the SWB1 data was more difficult to recognise and hence more confusable. This inference is reinforced by the fact that a narrower pruning beamwidth had to be used to reduce phone-lattices down to sizes comparable to those in the TIMIT evaluations, demonstrating that there were more lattice paths with high likelihoods.

Overall the DMPLS systems achieved more favourable performances compared to those obtained by the baseline HMM-based HMMS and lattice-based PLS systems for the SWB1 data. The DMPLS systems yielded considerably lower miss rates than PLS as well as significantly lower FA/kw and CPU/kw-hr rates than HMMS. The DMPLS system with the best compromise across all performance metrics was the DMPLS[5,10,100,2] system, yielding a low 13.9% miss rate, 36.1 FA/kw rate and a very low 0.18 CPU/kw-hr (1 hour processed in 10 seconds) execution time.

Method	Miss Rate	FA/ kw	CPU/ kw-hr
HMMS[-7500]	8.0	366.9	1.77
HMMS[-7300]	14.1	319.6	1.77
PLS[3,10,200,0]	38.4	3.2	-.–
DMPLS[3,10,200,2]	17.5	59.0	0.51
DMPLS[5,10,150,2]	11.0	83.6	0.72
DMPLS[5,10,150,1]	14.2	23.0	0.72
DMPLS[5,10,100,2]	13.9	36.1	0.18

Table 5. Keyword spotting results on SWB1

6. CONCLUSION

The reported experiments show that the Dynamic Match Phone-Lattice method provides very fast keyword spotting while maintaining respectable miss rate performance. In evaluations conducted on the Switchboard-1 database, DMPLS achieved a 13.9% miss rate and a 36.1 FA/kw rate, and was able to search 1 hour of speech in 10 seconds. This was a commendable result since baseline HMM-based HMMS system yielded only a slightly lower 8.0% miss rate, a significantly higher 366.9 FA/kw rate, and was 10 times slower, taking 1 minute 46 seconds to search an hour of speech. The considerable speed benefit of DMPLS and significantly lower FA/kw rate would more than compensate for a slightly increased miss rate in many applications. Additionally the use of MED optimisations would yield even faster speeds (a 35% relative speed improvement using the approach detailed in section 3.2) without degrading miss and FA/kw rates. Overall the experiments confirm that DMPLS is well suited for keyword spotting tasks that require very fast and accurate queries.

7. REFERENCES

[1] J. R. Rohlicek, *Modern methods of Speech Processing*, chapter Word Spotting, pp. 136–140, Kluwer Publishers, 1995.

[2] S. J. Young, M. G. Brown, J. T. Foote, G. J. F. Jones, and K. Sparck Jones, "Acoustic indexing for multimedia retrieval and browsing," in *IEEE International Conference on Acoustics, Speech and Signal Processing 1997*, 1997.

[3] J. G. Wilpon, L. R. Rabiner, C. H. Lee, and E. R. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden markov models," *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1990.

[4] M. J. H. Jurafsky, *Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition*, chapter Minimum Edit Distance, Prentice Hall, Upper Saddle River, 2000.

[5] D. A. James and S. J. Young, "A fast lattice-based approach to vocabulary independent wordspotting," in *IEEE International Conference on Acoustics, Speech and Signal Processing 1994*, Adelaide, Australia, 1994, vol. 1, pp. 377–380.

[6] K. Thambiratnam and S. Sridharan, "Isolated word verification using cohort word-level verification," in *Proceedings of Eurospeech 2003*, Geneva, Switzerland, 2003.