Rapid Yet Accurate Speech Indexing Using Dynamic Match Lattice Spotting

Kishan Thambiratnam, *Member, IEEE*, and Sridha Sridharan, *Senior Member, IEEE*

***Abstract*—The support for typically out-of-vocabulary query terms such as names, acronyms, and foreign words is an important requirement of many speech indexing applications. However, to date many unrestricted vocabulary indexing systems have struggled to provide a balance between good detection rate and fast query speeds. This paper presents a fast and accurate unrestricted vocabulary speech indexing technique named Dynamic Match Lattice Spotting (DMLS). The proposed method augments the conventional lattice spotting technique with dynamic sequence matching, together with a number of other novel algorithmic enhancements, to obtain a system that is capable of searching hours of speech in seconds while maintaining excellent detection performance.**

***Index Terms*—Audio indexing, information retrieval, keyword spotting, speech data mining, speech indexing.**

I. Introduction

THE growing importance of speech and multimedia data in society has necessitated the development of technologies that can index and search these mediums effectively. To date, the most common solution to this problem has been to use speech-to-text transcription (STT) systems to generate textual transcriptions that can then be rapidly searched using conventional text search engines (e.g., [1]–[3]). Unfortunately, such approaches are severely restricted by the vocabulary of the STT system, and thus are inappropriate for tasks that require support for typically out-of-vocabulary keyword queries such as names, acronyms, and foreign keywords. These tasks include news-story indexing, technical document database searching, and multilanguage surveillance.

For such applications, unrestricted vocabulary keyword spotting methods, such as HMM-based keyword spotting [4], have provided a solution, though at the expense of considerably slower query speeds. Faster approaches such as reverse dictionary lookup search methods [5] and lattice spotting techniques [6]–[9] offer significantly quicker searching but are encumbered by poor miss rate performance.

This paper presents and extends the work on the Dynamic Match Lattice Spotting (DMLS) method first presented in [10]. The DMLS method combines the fast performance of lattice spotting with dynamic sequence matching techniques to obtain more satisfactory keyword detection performance. The resulting system is capable of searching 1 h of audio in under 2 s while maintaining good detection performance.

Initial sections of this paper discuss the motivation for the DMLS method and present a detailed description of all associated algorithms. Subsequent sections then report on experiments to compare the performance of DMLS to other indexing methods. These sections also provide a detailed analysis of the various parameters of DMLS. The final sections discuss methods of optimising the execution speed of DMLS to improve real-time speed without affecting detection performance.

II. Motivation

Preliminary experiments using HMM-based keyword spotting found that although such systems provided good detection performance, they left much wanting in terms of execution speed. Specifically, it was found that it took 110 s [1] to search 1 h of speech for a single keyword using a 3-GHz Pentium 4 processor. Although such speeds are more than adequate for real-time monitoring tasks such as broadcast monitoring and keyword control systems, significantly faster speeds are required for tasks such as large database searching.

One solution is to first transcribe the audio using a STT system and then only search the resulting text at query time. Such an approach provides very fast searches since query time processing is purely textual. However, keyword queries using this method are restricted by the vocabulary of the STT system. In very large vocabulary domains or domains with dynamic vocabulary sets—such as the broadcast domain—this restriction will be problematic. For example, the name of the latest elected president of Sri Lanka is unlikely to be in the vocabulary of most STT systems, but may be of interest to a user searching for any related news stories.

In contrast, lattice-based and bottom-up indexing methods provide unrestricted query-time vocabularies while maintaining fast query speeds. Instead of transcribing the speech into words during the preparation stage, these methods use a low-level representation such as phone or syllable labels. This low-level representation can then be searched at query time to infer putative locations of a target word.

Unfortunately, the detection performances of such methods are significantly poorer than HMM-based methods. For example, the bottom-up method proposed by [5] achieved a miss rate of approximately 35% @ 10 FA/kw-h (false alarms per queried keyword per hour of speech searched). A miss rate of 30% @ 10 FA/kw-h was reported by [6] for lattice-based spotting.

[1]This timing was obtained using a HMM keyword spotter with a Gaussian mixture model (GMM) background model, as described by [11].

Manuscript received June 1, 2005; revised October 6, 2005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dilek Hakkani-Tur.

The authors are with the Speech Research Laboratory, Queensland University of Technology, Brisbane, QLD 4113, Australia (e-mail: k.thambi@gmail.com; s.sridharan@qut.edu.au).

Digital Object Identifier 10.1109/TASL.2006.872615

A major reason for the poor performance of lattice-based and bottom-up methods is that query time searching is based on highly erroneous phone recognizer transcriptions. Phone recognizer error rates are typically in the vicinity of 30%–50% in favorable conditions, and potentially even poorer for adverse conditions. This high error rate will clearly be propagated to the query-time search stages resulting in poor search performance.

Lattice-based approaches (e.g., [6]) attempt to accommodate phone recognizer errors by encoding an utterance within the recognition lattice in terms of multiple hypotheses. A phone lattice not only represents multiple utterance level transcriptions, but also maintains multiple localized transcriptions at a given time point in an utterance. It is hoped then, that at least one of the localized hypotheses occurring at the point of a true target keyword occurrence will match the target keyword phone sequence.

One means of further reducing the impact of phone recognizer error is to incorporate any prior knowledge of recognizer errors into the search process. For example, if the phones $/aa/$ and $/ih/$ are known to be highly substitutionary for a given phone recognizer, then improved detection rates may be obtained by including this prior information in the search process.

However, an unfortunate side-effect is that allowing for such error corrections will inadvertently lead to an increase in false alarm rates. For example, when using the $/aa/ \leftrightarrow /ih/$ substitution rule, true occurrences of the word $STICK = (/s/, /t/, /ih/, /k/)$ will be labeled as instances of the word $STOCK = (/s/, /t/, /aa/, /k/)$. As such, any error correction will need to incur some kind of cost that affects the overall likelihood of a putative instance.

A method that successfully incorporates phone recognizer error correction will improve overall keyword spotting robustness. The resultant gains in performance will improve the suitability of lattice-based and bottom-up approaches for the speech indexing task.

III. DMLS Method

DMLS is an extension of conventional lattice-based spotting, but uses the minimum edit distance (MED) [12] during lattice searching to compensate for phone recognizer insertion, deletion, and substitution errors. This addresses a major shortcoming of lattice-based methods—the requirement for the target phone sequence to appear in its entirety within the phone-lattice for consideration as a hypothesised keyword occurrence.

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 the DMLS method, each observed lattice phone sequence is scored against the target phone sequence using the MED. Lattice sequences are then accepted or rejected by thresholding on this MED score, hence providing robustness against phone recognizer errors. Conventional lattice-based indexing is a special case of DMLS where a score threshold of 0 is used.

This means of phone recognizer error correction has significant potential for improving detection performance. Consider the phone lattice segment shown in Fig. 1 corresponding to an instance of the word $STOCK$. Using the conventional lattice-based search, none of the paths will match the target sequence $STOCK = (/s/, /t/, /aa/, /k/)$. However, the lattice shows that the phone recognizer correctly transcribed three of the four phones correctly: $/s/$, $/t/$ and $/k/$. Yet a simple substitution error for the phone $/aa/$ prevents the word $STOCK$ from being detected. In contrast, the DMLS search will match a large number of paths, though each path will have a nonzero MED score. The decision to accept or reject these putative occurrences is then left to a subsequent thresholding or keyword verification stage (e.g., [13]). This is one of the many examples where DMLS' error robustness aids detection performance.

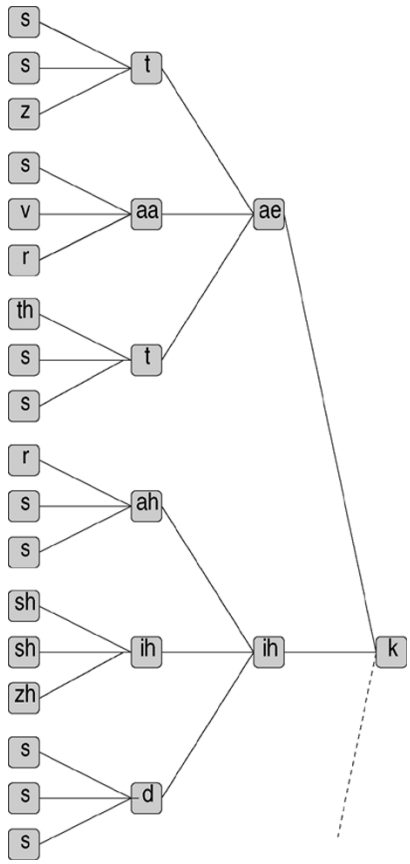


Fig. 1. Segment of phone lattice for an instance of the word $STOCK$.

As stated before, a downfall of DMLS is that there will be an increase in false alarm rate since the sequence matching process is significantly more *liberal* than the conventional lattice-based technique. However, since each putative occurrence will have an associated MED score that is indicative of the *looseness* of the match, simple techniques such as thresholding can be used to limit the *looseness* of the final result set. Additionally a subsequent keyword verification stage can be used to post-process the DMLS result set to further aid in false alarm reduction. Thus, although the anticipated increase in false alarm rate for the DMLS technique is unfortunate, it is an addressable issue.

Another dynamic sequence matching-inspired approach was previously proposed in [9]. This method differs from DMLS as it applies a dynamic programming match on a path/utterance level. In contrast, the proposed DMLS technique performs sequence matching on a localized per-word-length scale. It is believed that the DMLS approach is better suited to the spotting

task as keyword spotters seek to discriminate keywords from nonkeywords on a localized scale.

A. Basic Method

The DMLS algorithm is an extension of the conventional lattice-based indexing method. This is a two stage process consisting of an initial lattice building stage and a subsequent query-time search stage.

During the lattice building stage, each utterance is decoded using a Viterbi phone recognizer to generate a recognition phone lattice. The quality and complexity of this lattice can be controlled be adjusting a variety of factors including the phone language model, the word insertion penalty, the grammar scale factor, and the number of phone classes.

Lattice building only needs to be performed once per utterance. Once a lattice is built, it can be used repeatedly in subsequent queries regardless of the query term.

The second stage of DMLS is the lattice search stage. This step is considerably faster than the initial lattice building stage since processing is purely textual. The process consists of a modified Viterbi traversal of the lattice that emits putative matches during traversal.

Let $P = (p_1, \ldots, p_N)$ 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 $Q = (q_1, \ldots, q_M)$ be defined as the observed phone sequence obtained by traversing the current token history backward M levels, where:
      i) $M = N + MAX(C_i) \times S_{\max}$;
      ii) C_i is the insertion MED cost function.
   b) let $S = BESTMED(Q, P, C_i, C_d, C_s)$, where:
      i C_d is the deletion MED cost function;
      ii C_s is the substitution MED cost function;
      iii $BESTMED(\ldots)$ returns the score of the first element in the last column of the MED cost matrix that is less than or equal to $S_{\max}$ (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.

B. Improved Dynamic Match Lattice Search

The DMLS method described previously will execute significantly faster than HMM-based keyword spotting. This is because all search-time processing is purely textual. However, a significant part of this search process is Viterbi decoding, which in itself is a computationally intensive task. It is in fact possible to remove this Viterbi lattice traversal from query-time processing, as described below. This results in significant increases in query speed.

Since the paths traversed through the lattice are independent of the query term (traversal is done purely by maximum likelihood), it is possible to perform the lattice traversal during the lattice building stage. Then it is only necessary to store the observed phone sequences at each node for searching at query-time.

Furthermore, if it is assumed that the maximum queried phone sequence length is fixed at $N_{\max}$ and the maximum sequence match score threshold is preset at $S_{\max}$, then it is only necessary to store observed phone sequences of length $M_{\max} = N_{\max} + MAX(C_i) \times S_{\max}$. Clearly, a larger value of $N_{\max}$ will provide support for longer query sequences, but will also result in an increase in storage requirements. Thus it is important to carefully select a value of $N_{\max}$ that supports sufficiently long query sequences without unnecessarily penalizing storage requirements to support rare queries of very long sequences.

This optimization results in a significant reduction in the complexity of query-time processing. Whereas in the basic DMLS approach, full Viterbi traversal was required, processing using this optimized approach is now a linear progression through a set of observed phone sequences.

The improved lattice building algorithm is then given as follows.

1) Construct the recognition lattice using the same approach as in the basic DMLS method.
2) Let $A = \{\}$, where A is the collection of observed phone sequences
3) For each node in the phone-lattice, where node list traversal is done in time-order:
   a) for each token in the top K scoring tokens in the current node:
      i) let $Q = (q_1, \ldots, q_{M_{\max}})$ be the observed phone sequence obtained by traversing the token history backward $M_{\max}$ levels;
      ii) append the sequence Q to the collection A, as well as any associated timing information.
   b) 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.
4) Store the observed phone sequence collection for subsequent searching.
5) The recognition lattice can now be discarded as it is no longer required for query-time searching.

This allows the considerably simpler query time search algorithm shown below.

1) Load the previously computed observed phone sequence collection for the current utterance.
2) For each member, Q of the collection of observation sequences, A:
   a) let $S = BESTMED(Q, P, C_i, C_d, C_s)$;
   b) emit Q as a putative occurrence if $S \leq S_{\max}$.

IV. Evaluation of DMLS Performance

Experiments were performed to compare the DMLS technique with conventional indexing approaches. This initial set of experiments were performed on the TIMIT clean microphone speech database to limit the task complexity. Later sections in this paper present experiments on a more difficult conversational telephone speech task.

A. Baseline Systems

An HMM-based keyword spotter using a high-order GMM background model (as described in [11] and [13]) was used for evaluating HMM-based keyword spotter performance. This baseline provides a comparison with standard state-of-the-art keyword spotting performances achieved for real-time keyword spotting.

The lattice-based baseline system was constructed using the method proposed by [6]. This algorithm was implemented by simply using a DMLS system with $S_{\max} = 0$. This in essence would result in only exact matches within the recognition lattice being emitted as putative occurrences, as required for conventional lattice-based spotting. Miss and false alarm rates obtained using this approach would be indicative of conventional lattice-based spotting performance. However, true execution times could not be measured for this baseline system, as a simulated system was being used rather than the true lattice-based system described by [6].

B. Evaluation Set

A keyword spotting evaluation set was constructed using speech taken from the TIMIT test database. The choice of query words was constrained to words that had six-phone-length pronunciations to reduce target word length dependent variability.

Approximately 1 h of TIMIT test speech (excluding the SA1 and SA2 utterances that are repeated for every speaker, so as to reduce test set bias) was labeled as evaluation speech. From this speech, 200 six-phone-length unique words were randomly chosen and labeled as query words. These query words appeared a total of 480 times in the evaluation speech.

C. Recognizer Parameters

Sixteen-mixture triphone HMM acoustic models and a 256-mixture GMM background model were trained on a 140-h subset of the Wall Street Journal 1 (WSJ1) database for use in the experiments. Additionally 2-gram and 4-gram phone-level language models were trained on the same section of WSJ1 for use during the lattice building stages of DMLS and the conventional lattice-based methods. The resulting acoustic models achieved a word error rate of 21.1% on a TIMIT evaluation set using a 6 K word list.

All speech was parameterized using perceptual linear prediction coefficient feature extraction and cepstral mean subtraction. In addition to 13 static cepstral coefficients (including the 0th coefficient), deltas and accelerations were computed to generate 39-dimension observation vectors.

D. Lattice Building

The following lattice building procedure, based on the optimized lattice building approach described in Section III-B, was used for these experiments.

1) Lattices were generated for each utterance by performing a U-token Viterbi decoding pass using the 2-gram phone-level language model.
2) The resulting lattices were expanded using the 4-gram phone-level language model.
3) Output likelihood lattice pruning was applied using a beam-width of W to reduce the complexity of the lattices. This essentially removed all paths from the lattice that had a total likelihood outside a beamwidth of W of the top-scoring path.
4) A second V-token traversal was performed to generate the top ten scoring observed phone sequences of length 11 at each node (allowing detection of sequences of up to $11 - MAX(C_i) \times S_{\max}$ phones).

Lattice building was only performed once per utterance. The resulting phone sequence collections were then stored to disk and used during subsequent query-time search experiments.

E. Query-Time Processing

The optimized lattice search algorithm described in Section III-B was used for these experiments. The sequence matching threshold $S_{\max}$ was fixed at 2 for all experiments unless noted otherwise. MED calculations used a constant deletion cost of $C_d = \infty$, as preliminary experiments obtained poor results when noninfinite values of C_d were used. The insertion cost was also fixed at $C_i = 1$.

In contrast, C_s was allowed to vary based on phone substitution rules. Initially, experiments attempted to use costs directly estimated from the phone confusion matrix. However, this introduced a considerable dependence of the cost function upon the actual recognizer characteristics, and although this dependence could be reduced through smoothing, it was decided to use a simpler set of knowledge-driven rules to provide a greater degree of generalization. Further experimentation found that there was no loss in performance using this set of knowledge-driven rules compared to a data-driven estimation from the phone confusion matrix, thus further reinforcing the choice to use a knowledge-derived rule set.

The basic rules used to obtain these costs are shown in Table I(a) and were determined by examining phone recognizer confusion matrices. However, some exceptions to these rules were made based on empirical observations from small scale experiments using a separate development data set. The final set of substitution costs used in the reported experiments are given in Table I(b). Substitutions were completely symmetric. Hence, the substitution of a phone m in a given phone group with another phone n in the same group yielded the same cost as the substitution of n with phone m.

F. Evaluation Procedure

The systems were evaluated by performing single-word keyword spotting for each query word across all utterances in the evaluation set. The total miss rate for all query words and the false alarm per queried word per hour (FA/kw-h) were then calculated using reference transcriptions of the evaluation data. Additionally the total CPU processing seconds per queried keyword per hour (CPU/kw-h) was measured for each experiment using a 3-GHz Pentium 4 processor.

For DMLS, CPU/kw-h only included the CPU time used during the DMLS search stage. That is, the time required for lattice building was not included.

All experiments used a commercial-grade decoder to ensure that the best possible CPU/kw-h results were reported for the HMM-based system. This is because HMM-based keyword spotting time performance is bounded by decoder performance.

TABLE I
PHONE SUBSTITUTION COSTS FOR DMLS

Rule	Cost
close consonant substitution eg. $/n/ \leftrightarrow /nx/$, $/z/ \leftrightarrow /zh/$	0
vowel substitutions	1
closure and stop substitutions	1
all other substitutions	∞

(a)

Phones	Cost	Phones	Cost
aa ae ah ao aw ax ay	1	d dh	0
eh en er ey ih iy		n nx	0
ow oy uh uw		t th	0
b d dh g k p t th jh	1	uw w	1
z zh s sh	1	w wh	0

(b)

TABLE II
BASELINE RESULTS EVALUATED ON TIMIT

Method	Miss Rate	FA/ kw-hr	CPU/ kw-hr
HMM[∞]	1.6	53.1	94.8
HMM[-7580]	10.4	43.9	94.8
HMM[-7000]	39.8	20.2	94.8
CLS[3,10,200,0]	32.9	0.5	-.–
DMLS[3,10,200,2]	10.2	22.2	18.0

TABLE III
COST RULES FOR ISOLATED RULE DMLS SYSTEMS

System	C_d	C_i	$C_s(a,b)$
Close consonant subst	∞	∞	a and b close consonants, 1 otherwise, ∞
Vowel subst	∞	∞	a and b are vowels, 1 otherwise, ∞
Closure/stop subst	∞	∞	a and b are closure/stop, 1 otherwise, ∞
Insertions	∞	1	∞

G. Results

To aid discussion, the notation DMLS$[U, V, W, S_{\max}]$ is used to specify DMLS 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 HMM$[\alpha]$ is used when referring to baseline HMM systems, where α was the duration-normalized output likelihood threshold used. Additionally, the baseline conventional lattice-based method is referred to as CLS.

Performances for the DMLS, HMM-based and lattice-based systems measured for the TIMIT evaluation set are shown in Table II. For this set of experiments, the DMLS[3,10,200,2] configuration was chosen as the baseline DMLS configuration (the decision was based on small scale experiments used to establish sensible but not necessarily optimal parameter values)

The timing results demonstrate that as expected DMLS was significantly faster than the HMM method, running at approximately five times the speed. This amounts to a baseline DMLS system capable of searching 1 h of speech in 18 s. DMLS also had more favorable FA/kw-h performance: at 10.2% miss rate, it had a FA/kw-h rate of 22.2, significantly lower than the 43.9 FA/kw-h rate achieved by the HMM$[-7580]$ system. However, the HMM system was still capable of achieving a much lower miss rate of 1.6% using the HMM$[\infty]$ configuration, though at the expense of considerably more false alarms.

The miss rate achieved by the conventional lattice-based system was very poor compared to that of DMLS. This confirms that the phone error robustness inherent in DMLS yields considerable detection performance benefits. However, the false alarm rate for CLS was dramatically better than all other systems, though with such a high miss rate, this is not surprising.

V. ANALYSIS OF DYNAMIC MATCH RULES

A notable result from the previous section was the considerable improvement in achievable miss rate for DMLS over the baseline lattice-based system. This indicated that the phone recognizer error robustness incorporated into the DMLS search did significantly improve detection performance. However, it was not immediately clear which aspects of the dynamic match process were most effective in improving performance.

Specifically, improvements in performance could be attributed to the four main cost rules used in the dynamic match process: insertions, close consonant substitutions (e.g., $/d/ \leftrightarrow /dh/$, $/n/ \leftrightarrow /nx/$), vowel substitutions and closure/stop substitutions (e.g., $/b/ \leftrightarrow /d/$, $/k/ \leftrightarrow /p/$). As such, experiments were performed to quantify the benefits of individual cost rules.

Specialized DMLS systems were built to evaluate the effects of individual cost rules in isolation. The systems were implemented by using customised MED cost functions shown in Table III. The evaluation set, recognizer parameters, experimental procedure, and DMLS algorithm are the same as used in Section IV.

A. Results

Table IV shows the results of the specialized DMLS systems, baseline lattice-based CLS system, and the previously evaluated DMLS[3,10,200,2] system with all MED rules.

The experiments demonstrate that the magnitude of contributions of the various rules to overall detection performance varies drastically. Interestingly no single rule brought performance down to the *all rules* DMLS system. This indicates that the rules are complementary in nature and yield a combined overall improvement in miss rate performance.

TABLE IV
TIMIT PERFORMANCE WHEN ISOLATING VARIOUS DP RULES

Method	Miss Rate	FA/ kw-hr
CLS[3,10,200,0]	32.9	0.5
DMLS[3,10,200,2] insertions	28.5	1.4
DMLS[3,10,200,2] close consonant subst	31.0	0.6
DMLS[3,10,200,2] vowel subst	15.6	9.4
DMLS[3,10,200,2] closure/stop subst	23.5	3.6
DMLS[3,10,200,2] all rules	10.2	22.2

Using the close consonant substitution rules only yielded a small gain in performance over the null-rule CLS system: 1.9% absolute in miss rate with only a 0.1 drop in FA/kw-h rate. The result suggests that the phone-lattice is already robust to close consonant substitutions, and as such, inclusion of this does not obtain significant gains in performance. Empirical study of the phone-lattices revealed this to be the case in many situations. For example, typically if the phone $/s/$ appeared in the lattice, then it was almost guaranteed that the phone $/sh/$ also appeared at a similar time location in the lattice.

The insertions-only system yielded a slightly larger gain of 4.4% absolute in miss rate with only a 0.9 drop in FA/kw-h rate. The result indicates that the lattices contain extraneous insertions across many of the multiple hypotheses paths, preventing detection of the target phone sequence when insertions are not accounted for. This observation is to be expected since phone recognizers typically do have significant insertion error rates, even when considering multiple levels of transcription hypotheses.

A significant absolute miss rate gain of 17.3% was observed for the vowel substitution system. However, this gain was at the expense of a 8.9 absolute increase in FA/kw-h rate. This is a pleasing gain and is supported by the fact that vowel substitution is a frequent occurrence in the realization of speech. As such, incorporating support for vowel substitutions in DMLS not only corrects errors in the phone recognizer but also accomodates this substitutionary habit of human speech.

Finally, significant gains were also observed for the closure/stop substitution system. An absolute gain of 9.4% in miss rate combined with an unfortunate 3.1 absolute increase in FA/kw-h rate was obtained for this system. Typically closures and stops are shorter acoustic units and, therefore, more likely to yield classification errors. As such, even though the phone lattice encodes multiple hypotheses, it appears that it is still necessary to incorporate robustness against closure/stop confusion for lattice-based keyword spotting.

Overall, the experiments demonstrate the benefits of the various classes of MED rules used in the evaluated DMLS systems. It was pleasing to note that even the simplest of these rules still provided tangible gains in performance over the baseline lattice-based CLS system. This clearly reinforces the fact that the dynamic matching aspects of DMLS are beneficial. The results showed that insertion and close consonant substitution rules only provided a small performance benefit over a conventional lattice-based system, whereas vowel and closure/stop substitution rules yielded considerable gains in miss rate. Gains in miss rate were typically unfortunately offset by increases in FA/kw-h rate, although the majority of these gains were fairly small, and would most likely be justifiable in light of the resulting improvements in miss rate.

TABLE V
EFFECT OF ADJUSTING NUMBER OF LATTICE GENERATION TOKENS

Method	Miss Rate	FA/kw-hr	CPU/kw-hr
DMLS[3,10,200,2]	10.2	22.2	18.0
DMLS[5,10,200,2]	5.8	46.0	42.6

TABLE VI
EFFECT OF ADJUSTING PRUNING BEAMWIDTH

Method	Miss Rate	FA/kw-hr	CPU/kw-hr
DMLS[3,10,150,2]	12.5	14.6	10.8
DMLS[3,10,200,2]	10.2	22.2	18.0
DMLS[3,10,250,2]	9.2	29.6	28.2
DMLS[3,10,∞,2]	7.3	72.7	175.8

VI. ANALYSIS OF DMLS ALGORITHM PARAMETERS

The experiments reported earlier in this paper used a fixed DMLS[3,10,200,2] configuration to reduce the scope of experiments. This section reports on experiments to quantify the effects of individual algorithm parameters on detection performance. The evaluation set, recognizer parameters, experimental procedure, and DMLS algorithm are the same as used in Section IV.

A. Number of Lattice Generation Tokens

The number of tokens used for lattice generation U has a direct impact on the maximum size of the resulting phone lattice. For example, if a value of $U = 3$ is used, then a lattice node can have at most three predecessor nodes. Whereas, if a value of $U = 5$ is used, then the same node can have up to five predecessor nodes, greatly increasing the size and complexity of the lattice when applied across all nodes.

Tuning of U directly affects the number of hypotheses encoded in the lattice, and hence the best achievable miss rate. However, using larger values of U also increases the number of nodes in the lattice, resulting in an increased amount of processing during DMLS searching and, therefore, increased execution time.

Table V shows the result of increasing U from 3 to 5. As expected, increasing U resulted in an improvement in miss rate of 4.4% absolute but also in an increase in execution time by a factor of 2.3. A corresponding 23.8 absolute increase in FA/kw-h rate was also observed.

The obvious benefit of tuning the number of lattice generation tokens is that appreciable gains in miss rate can be obtained. Although this has a negative effect on FA/kw-h rate, a subsequent keyword verification stage may be able to accommodate the increase.

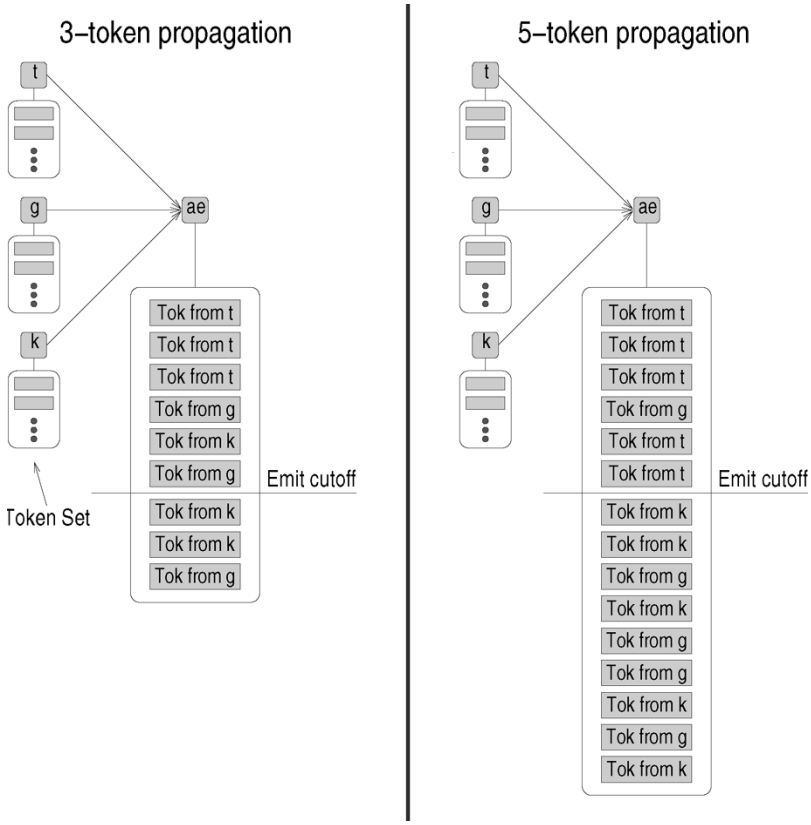


Fig. 2. Effect of lattice traversal token parameter.

B. Pruning Beamwidth

Lattice pruning is applied to remove less likely paths from the generated phone lattice, thus making the lattice more compact. This is typically necessary when language model expansion is applied. For example, applying 4-gram language model expansion to a lattice generated using a 2-gram language model results in a significant increase in the number of nodes in the lattice, many of which may now have much poorer likelihoods due to additional 4-gram language model scores.

The direct benefit of applying lattice pruning is an immediate reduction in the size of the lattice that needs to be searched. This will give improvements in execution time, though at the expense of losing potentially correct paths that unfortunately did not score well linguistically.

Table VI shows the effect of pruning beamwidth for four different values: 150, 200, 250, and ∞. As predicted, decreasing pruning beamwidth yielded significant gains in execution speed at the expense of reductions in miss rate. Corresponding drops in FA/kw-h rate were also observed.

Adjusting pruning beamwidth appears to be particularly well suited for tuning execution time. The changes in CPU/kw-h figures were dramatic, and in comparison, the miss rate figures varied in a much smaller range.

C. Number of Lattice Traversal Tokens

The number of lattice traversal tokens V corresponds to the number of tokens used during the secondary Viterbi traversal. Tuning this parameter affects how many tokens are propagated out from a node, and hence, the number of paths entering a node that survive subsequent propagation.

TABLE VII
EFFECT OF ADJUSTING NUMBER OF TRAVERSAL TOKENS

Method	Miss Rate	FA/kw-hr	CPU/kw-hr
DMLS[3,5,200,2]	10.4	20.9	16.8
DMLS[3,10,200,2]	10.2	22.2	18.0
DMLS[3,20,200,2]	9.8	22.6	17.4

The impact of this on DMLS is actually more subtle, and is demonstrated by Fig. 2. In this instance, the scores of tokens propagated from the t node are much higher than the scores from the other nodes. As such, in the five-token propagation case, the majority of the high-scoring tokens in the target node are from the t node. Hence, the tokens above the emission cutoff (i.e., the tokens from which observed phone sequences are generated) are mainly t nodes. However, using the same emission cutoff and three-token propagation results in a set of top-scoring tokens from a variety of source nodes. It is not immediately obvious whether it is better to use a high or low number of lattice traversal tokens for optimal DMLS performance.

Table VII shows the results of experiments using three different numbers of traversal tokens: 5, 10, and 20. It appears that all three measured performance metrics were fairly insensitive to changes in the number of traversal tokens. There was a slight decrease in miss rate when using a higher value of V, though this may not be considered a dramatic enough change to justify the additional processing burden required at the lattice building stage.

TABLE VIII
EFFECT OF ADJUSTING MED COST THRESHOLD $S_{\max}$

Method	Miss Rate	FA/kw-hr	CPU/kw-hr
DMLS[3,10,200,0]	31.0	0.6	18.0
DMLS[3,10,200,1]	13.3	5.2	18.0
DMLS[3,10,200,2]	10.2	22.2	18.0
DMLS[3,10,200,3]	8.7	62.4	18.0

D. MED Cost Threshold

Tuning of the MED cost threshold, $S_{\max}$, is the most direct means of tuning miss and FA/kw-h performance. However, if discrete MED costs are used, then $S_{\max}$ itself will be a discrete variable, and as such, thresholding will not be on a continuous scale.

The $S_{\max}$ parameter controls the maximum allowable discrepancy between an observed phone sequence and the target phone sequence. Experiments were carried out to study the effects of changes in $S_{\max}$ on performance. The results of these experiments are shown in Table VIII. Since thresholding was applied on the result set of DMLS, there were no changes in execution time.

The experiments demonstrated that adjusting $S_{\max}$ gave dramatic changes in FA/kw-h. In contrast, the changes in miss rate were considerably more conservative except for the $S_{\max} = 0$ case. Tuning of the MED cost threshold therefore appears to be most applicable to adjusting the FA/kw-h operating point. This is intuitive since adjusting $S_{\max}$ adjusts how much error an observed phone sequence is allowed to have, and as such has a direct correlation with false alarm rate.

E. Tuned Systems

Previous sections examined tuning of the various DMLS parameters in isolation. However, it was not clear from these experiments how a system constructed using a combination of tuned parameters would perform. In particular, it was essential to know whether the benefits obtained from tuning the individual parameters were complementary, resulting in even greater increases in keyword spotting performance.

As such, two tuned systems were constructed and evaluated on the TIMIT data set. Parameters for these systems were selected as follows.

1) The number of lattice generation tokens appeared to be well suited to adjusting miss rate. As such, a value of $U = 5$ was used for the tuned systems.
2) DMLS performance appeared insensitive to changes in the number of lattice traversal tokens. Hence, to remain consistent with previous experiments, a value of $V = 10$ was used.
3) The speed increases observed using a reduced lattice pruning beamwidth were quite dramatic and in comparison only resulted in a small decrease in miss rate. Considering the anticipated gains in miss rate from the increase in the number of lattice generation tokens, a reduced value of $W = 150$ was used.
4) Two values of $S_{\max}$ were evaluated to obtain performance at different false alarm points. The values evaluated were $S_{\max} = 1$ and $S_{\max} = 2$. Although it was anticipated that a reduction in miss rate would be observed for the lower $S_{\max} = 1$ system, it was hoped that this would be compensated for by the increase in the number of lattice generation tokens, and further justified by the significantly lower false alarm rate.

TABLE IX
TUNED DMLS CONFIGURATIONS EVALUATED ON TIMIT

Method	Miss Rate	FA/kw-hr	CPU/kw-hr
Untuned DMLS[3,10,200,2]	10.2	22.2	18.0
Tuned DMLS[5,10,150,1]	11.5	6.7	18.6
Tuned DMLS[5,10,150,2]	7.3	26.8	18.6

The results of the tuned systems on the TIMIT evaluation set are shown in Table IX. The first system achieved a significant reduction in FA/kw-h rate over the initial DMLS[3,10,200,2] system at the expense of only a small 1.3% absolute increase in miss rate. The second system obtained a good decrease in miss rate of 2.9% with only a small 4.6 FA/kw-h rate increase. Both these systems maintained almost the same execution speed as the initial DMLS system. It is difficult to say which of the tuned systems is more optimal, since typically the choice of operating point is application dependent.

VII. CONVERSATIONAL TELEPHONE SPEECH EXPERIMENTS

Previous sections of this paper only evaluated DMLS on the clean microphone speech domain. The conversational telephone speech domain is a more difficult domain but is more representative of a real-world practical application of DMLS. As such, this section reports on experiments to evaluate the performance of DMLS for this domain.

Specifically, experiments were performed using the Switchboard 1 telephone speech corpus. To maintain consistency, the same baseline systems, DMLS algorithms and evaluation procedure as used in previous sections were used here.

A. Evaluation Set and Recognizer Parameters

The evaluation set was constructed in a similar fashion to the previously constructed TIMIT evaluation set. Approximately two hours of speech was taken from the Switchboard corpus and labeled as evaluation speech. From this speech, 360 six-phone-length unique words were randomly chosen and marked as query words. In total, these query words appeared a total of 808 times in the evaluation set.

Acoustic and language models were trained up on a 165-h subset of the Switchboard-1 database using the same approach as used for the previous TIMIT experiments. The resulting acoustic models achieved a word error rate of 31.1% on a 6-h Switchboard-1 evaluation set using a 20 K word list.

B. Results

The results for the HMM-based, conventional lattice-based, and DMLS experiments on conversational speech SWB1 data are shown in Table X. DMLS performance was measured using the baseline DMLS[3,10,200,2] system as well as a number of tuned configurations. Tuned systems were constructed using a

TABLE X
KEYWORD SPOTTING RESULTS ON SWB1

Method	Miss Rate	FA/kw-hr	CPU/kw-hr
HMM[-7500]	8.0	410.9	106.2
HMM[-7300]	14.1	358.0	106.2
CLS[3,10,200,0]	38.4	3.6	-.–
DMLS[3,10,200,2]	17.5	66.1	30.6
DMLS[5,10,150,2]	11.0	93.6	43.2
DMLS[5,10,150,1]	14.2	25.8	43.2
DMLS[5,10,100,2]	13.9	40.4	10.8

		a	d	c	k	d
	0	1	2	3	4	5
a	1	0	1	2	3	4
b	2	1	1	2	3	4
c	3	2	2	1	2	3
d	4	3	2	2	2	2

(a)

		a	d	c	k	d	e
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
b	2	1	1	2	3	4	5
c	3	2	2	1	2	3	4
d	4	3	2	2	2	2	3

(b)

Fig. 3. Relationship between cost matrices for subsequences. (a) $\Omega(A, B')$. (b) $\Omega(A, B)$.

combination of lattice generation tokens, pruning beamwidth, and $S_{\max}$ tuning.

Of note is the dramatic increase in FA/kw-h rates for all systems compared to those observed for the TIMIT evaluations. This is an expected result, since the conversational telephone speech domain is a more difficult domain for recognition. For DMLS, this increase in false alarm rate is a result of the increased complexity of the lattices. It was found that the lattices generated for the Switchboard data were significantly larger than those generated for the TIMIT data when using the same pruning beamwidth. This meant that there were more paths with high likelihoods, indicating a greater degree of confusability within the lattices. As a result, more false alarms were generated.

Losses in miss rate in the vicinity of 5% absolute were also observed for all systems compared to the TIMIT evaluations. Although this is unfortunate, these losses are still minor in light of the increased difficulty of the data.

Overall though, DMLS still achieved more favorable performance than the baseline HMM-based and lattice-based systems. The DMLS systems not only yielded considerably lower miss rates than CLS but also significantly lower FA/kw-h and CPU/kw-h rates than the HMM-based systems.

In terms of detection performance, the two best DMLS systems were the DMLS[5,10,150,1] and the DMLS[5,10,100,2] configurations. Both had lower false alarm rates than the other DMLS systems and still maintained fairly low miss rates. However, the execution speed of the DMLS[5,10,100,2] configuration was four times faster than the DMLS[5,10,150,1] system. In fact, this system was capable of searching 1 h of speech in 10 s and thus would be more appropriate for applications requiring very fast search speeds.

Overall, the experiments demonstrate that DMLS is capable of delivering good keyword spotting performance on the more difficult conversational telephone speech domain. Although there was some degradation in performance compared to the clean speech microphone domain, the losses were in line with what would be expected. Also, DMLS offered much faster performance than the HMM-based system and considerably lower miss rates than the conventional lattice-based system.

VIII. NONDESTRUCTIVE OPTIMIZATIONS

Experiments on the TIMIT and Switchboard databases have clearly demonstrated that DMLS is capable of obtaining very fast keyword spotting speeds. Although these speeds are impressive, further gains in throughput can be obtained through optimization of the MED calculations.

MED calculations are in fact the mostly costly operations performed during the DMLS search stage. The basic MED algorithm is an $\mathrm{O}(N^2)$ algorithm and, hence, not particularly suitable for high-speed calculation. However, within the DMLS search context, two specific optimizations can be applied to reduce the computational cost of these MED calculations. These optimizations are the prefix sequence optimization and the early stopping optimization.

A. Prefix Sequence Optimization

The prefix sequence optimization utilizes the similarities in the MED cost matrix of two observed phone sequences that share a common prefix sequence.

Let $A = (a_1, a_2, \ldots, a_N)$, and $B = (b_1, b_2, \ldots, b_M)$. Also, let B' be defined as the first-order prefix sequence of B, given by $B' = (b_i)_1^{M-1}$. Finally, let the MED cost matrix between two sequences be defined as $\Omega(X, Y)$.

From the basic definition of the MED cost matrix, the $(N + 1) \times M$ cost matrix $\Omega(A, B')$ is equal to the first M columns of the cost matrix $\Omega(A, B)$. This is because B' is equal to the first $M - 1$ elements of B.

Therefore, given the cost matrix, $\Omega(A, B')$, it is only necessary to calculate the values of the $(M+1)$th column of $\Omega(A, B)$ to obtain the full cost matrix $\Omega(A, B)$. This is demonstrated in Fig. 3.

The argument extends to even shorter prefix sequences of B. For example, let B''' be defined as the third-order prefix sequence of B, given by $B''' = (b_i)_1^{M-3}$. Then, given $\Omega(A, B''')$, it is only necessary to calculate the $(M-1)$th, Mth, and $(M+1)$th column of $\Omega(A, B)$ to obtain the full cost matrix.

Now, given that the MED cost matrix $\Omega(A, B)$ is known, consider the task of calculating the MED cost matrix $\Omega(A, C)$. Let $P(B, C)$ return the longest prefix sequence of B that is also a prefix sequence of C. Then, $\Omega(A, C)$ can be obtained by taking $\Omega(A, B)$ and recalculating the last $|C| - |P(B, C)|$ columns.

In DMLS, an utterance is represented by a collection of observed phone sequences. Typically, there is a degree of prefix similarity between sequences from the same temporal location, and in particular between sequences emitted from the same node. As demonstrated above, knowledge of prefix similarity will allow a significant reduction in the number of MED calculations required.

Observed Phone Sequences	Prefix Similarity	# MED cols to calculate
aa b ae g k b ae	0	7
aa b ae g k b eh	6	1
aa b ae g k d ae	5	2
aa b ae k t ow wh	3	4
aa b ae k t uh ae	5	2
aa g eh sh hh ow p	1	6

Fig. 4. Demonstration of the MED prefix optimization algorithm.

The simplest means of obtaining this knowledge is to simply sort the phone sequences of an utterance lexically during the lattice building stage. Then, the degree of prefix similarity between each sequence and its predecessor can be calculated and stored. For this purpose, the degree of prefix similarity is defined as the length of the longest common prefix subsequence of two sequences.

Then, during the DMLS search stage, all that is required is to step through the sequence collection and use the predetermined prefix similarity value to determine what portion of the MED cost matrix needs to be calculated, as demonstrated in Fig. 4. As such, only changed portions of the MED cost matrix are iteratively updated, greatly reducing computational burden.

B. Early Stopping Optimization

The early stopping optimization uses knowledge about the $S_{\max}$ threshold to limit the extent of the MED matrix that has to be calculated.

From MED theory, the element $\Omega(X,Y)_{i,j}$ of the MED cost matrix $\Omega(X,Y)$ corresponds to the minimum cost of transforming the sequence $(x)_1^i$ to the sequence $(y)_1^j$. For convenience, the notation Ω is used to represent $\Omega(X,Y)$. The value of $\Omega_{i,j}$ is given by the recursive expression

$$\Omega_{i,j} = Min \begin{pmatrix} \Omega_{i-1,j-1} & + & C_s(x_i,y_j), \\ \Omega_{i-1,j} & + & C_d(x_i), \\ \Omega_{i,j-1} & + & C_i(y_j) \end{pmatrix}. \quad (1)$$

Given the above formulation, and assuming nonnegative cost functions, the value of $\Omega_{i,j}$ has a lower bound (LB) governed by

$$LB(\Omega_{i,j}) \geq Min\left(\{\Omega_{k,j}\}_{k=1}^{i-1} \cup \{\Omega_{k,j-1}\}_{k=1}^{|X|}\right). \quad (2)$$

That is, it is bounded by the minimum value of column $j-1$ and all values above row i in column j. This states that the lower bound of $\Omega_{i,j}$ is a function of $\Omega_{i-1,j}$, which implies the recursive formulation

$$LB(\Omega_{i,j}) \geq Min\left(\{LB(\Omega_{i-1,j})\} \cup \{\Omega_{k,j-1}\}_{k=1}^{|X|}\right). \quad (3)$$

This states that the lower bound of $\Omega_{i,j}$ is governed by all entries in the previous column and the lower bound of the element directly above it in the cost matrix. If the recursion is continuously unrolled, then the lower bound reduces to being only a function of the previous column and the very first element in column j, that is

$$LB(\Omega_{i,j}) \geq Min\left(\{LB(\Omega_{1,j})\} \cup \{\Omega_{k,j-1}\}_{k=1}^{|X|}\right). \quad (4)$$

Now, MED theory states that $\Omega_{1,j} = j \times C_i(y_j)$ for all values of j. This means that for a positive insertion cost function

$$LB(\Omega_{1,j}) \geq LB(\Omega_{1,j-1}). \quad (5)$$

Substituting this back into (4) gives

$$LB(\Omega_{i,j}) \geq Min\left(\{LB(\Omega_{1,j-1})\} \cup \{\Omega_{k,j-1}\}_{k=1}^{|X|}\right). \quad (6)$$

This reduces to the simple relationship

$$LB(\Omega_{i,j}) \geq Min\left(\{\Omega_{k,j-1}\}_{k=1}^{|X|}\right). \quad (7)$$

It has, therefore, been demonstrated that the lower bound of $\Omega_{i,j}$ is only a function of the values of the previous column of the MED matrix. This lends itself to a significant optimization within the DMLS framework.

Since $S_{\max}$ is fixed prior to the DMLS search, there is an upper bound on the MED score of observed phone sequences that are to be considered as putative hits. When calculating columns of the MED matrix, the relationship in (7) can be used to predict what the lower bound of the current column is. If this lower bound exceeds $S_{\max}$, then it is not necessary to calculate the current or any subsequent columns of the cost matrix, since all elements will exceed $S_{\max}$.

This is a very powerful optimization, particularly when comparing two sequences that are very different. It means that, in many cases, only the first few columns will need to be calculated before it can be declared that a sequence is not a putative occurrence.

C. Combining Optimizations

The early stopping optimization and the prefix sequence optimization can be easily combined to give even greater speed improvements. Essentially, the prefix sequence optimization uses prior information to eliminate computation of the starting columns of the cost matrix, while the early stopping optimization uses prior information to prevent unnecessary computation of the final columns of the cost matrix.

When combined, all that remains during MED costing is to calculate the necessary in-between columns of the cost matrix. As such, the combined algorithm is given as follows.

1) Initialize a MED cost matrix of size $(N+1) \times (M+1)$, where N is the length of the target phone sequence, and M is the maximum length of the observed phone sequences.
2) For each sequence in the observed phone sequence collection:
   a) Let k be defined as the previously computed degree of prefix similarity metric between this sequence and the previous sequence.

b) Using the prefix sequence optimization, it is only necessary to update the trailing columns of the MED matrix. Thus, for each column j from $(M+1)-k+1$ to $M+1$ of the MED cost matrix:
   i) Determine the minimum score, $MinScore(j-1)$ in column $j-1$ of the cost matrix.
   ii) If $MinScore(j-1) > S_{\max}$, then using the early stopping optimization, this sequence can be declared as not being a putative occurrence and processing can stop.
   iii) Calculate all elements for column j of the cost matrix.

c) Obtain $S = BESTMED(\ldots)$ in the normal fashion given this MED cost matrix.

IX. Optimized System Timings

Experiments were performed to evaluate the execution time benefits of the prefix sequence and early stopping optimizations. Five systems were evaluated.

1) **NOPT:** DMLS system without prefix sequence and early stopping optimizations.
2) **ESOPT:** DMLS system with early stopping optimization.
3) **PSOPT:** DMLS system with prefix sequence optimization.
4) **COPT:** DMLS system with combined early stopping and prefix sequence optimizations.
5) **CXOPT:** COPT system with miscellaneous coding optimizations applied such as removal of dynamic memory allocation, more efficient passing of data, etc.

A. Experimental Procedure

Experiments were performed using 100 randomly selected utterances from the Switchboard evaluation set detailed in Section VII-A. Single-word keyword spotting was performed for each utterance using a six-phone-length target word.

Each utterance was processed repeatedly for the same word 1400 times, and the total execution time was measured for all passes. The total time was then summed across all tested utterances to obtain the total time required to perform 100×1400 passes.

The relative speeds were calculated by finding the ratio between the measured speed of the tested system and the measured speed of the baseline NOPT system. The entire evaluation was then repeated a total of ten times, and the average relative speed factor was calculated. Execution time was measured on a single 3-GHz Pentium 4 processor.

Additionally, the final putative occurrence result sets were examined to ensure that exactly the same miss rate and FA/kw-h rates were obtained across all methods, since both optimizations should not affect these metrics.

B. Results

Table XI shows the speed of each system relative to the baseline unoptimized NOPT system. Tests were performed using $S_{\max}$ values of 2 and 4, since the benefits of the early stopping optimization depend on the value of $S_{\max}$.

The results clearly demonstrate that both optimizations yielded significant speed benefits. An even more pleasing result was that the two optimizations combined effectively to reduce execution time by a factor of 4 for the $S_{\max} = 2$ tests, and by a factor of 3 for the $S_{\max} = 4$ tests. Overall the fully optimized CXOPT system ran about five to six times faster than the original unoptimized system.

TABLE XI
Relative Speeds of Optimized DMLS Systems

S_{max}	System	Relative speed factor
2	NOPT	1.00
2	PSOPT	0.61
2	ESOPT	0.42
2	COPT	0.25
2	CXOPT	0.16
4	NOPT	1.00
4	PSOPT	0.60
4	ESOPT	0.63
4	COPT	0.32
4	CXOPT	0.21

TABLE XII
Fully Optimized System on Switchboard

Method	Miss Rate	FA/ kw-hr	CPU/ kw-hr
DMLS[5,10,100,2]	13.9	40.4	10.8
DMLS[5,10,100,2] with CXOPT	13.9	40.4	1.8

Table XII shows the execution time of the unoptimized DMLS system evaluated in Section VII as well as the CPU/kw-h figure for the same system incorporating the early stopping and prefix sequence optimizations. It can be seen that the resultant speed is 1.8 CPU/kw-h. This is an impressive result and clearly emphasises the suitability of DMLS for very fast large database keyword spotting applications.

X. Conclusion

This paper has presented a novel unrestricted vocabulary speech document indexing method named dynamic match lattice spotting. Through experimentation, it was demonstrated that the technique was capable of searching hours of data using only seconds of processing time, while maintaining excellent detection performance for both the conversational telephone and microphone speech domains.

The lack of robustness to subevent recognizer error was identified as a reason for the poor detection performance of preexisting indexing techniques. It was postulated that incorporating prior knowledge of subevent recognizer errors would be a means of improving detection rates. The DMLS method was proposed as a means of doing this.

Initial experiments using DMLS demonstrated that it outperformed preexisting techniques for the clean microphone speech domain. Compared to HMM-based keyword spotting, DMLS was significantly faster and also obtained considerably lower false alarm rates. Comparisons with the conventional lattice-based technique demonstrated the miss rate performance of DMLS to be vastly superior.

An analysis of the contributions of dynamic matching rules to DMLS performance was presented, to rationalize the benefits of DMLS over the conventional lattice search. It was found that the vowel substitution and closure/stop substitution rules contributed significantly to improving miss rate performance, while the close consonant substitution and insertion rules only offered small improvements. Nevertheless, in all cases, inclusion of any given dynamic matching rule offered clear benefits over the null-rule conventional lattice-based method.

A study of key parameters of DMLS was also presented. It was found that tuning of these parameters offered the ability to significantly enhance DMLS performance. Through careful adjustment of these parameters, it was possible to construct a tuned DMLS system that outperformed the previously evaluated baseline DMLS system.

Evaluation results were also provided for the conversational telephone speech domain. As would be expected, there was some degradation in performance compared to the clean microphone speech domain. Nevertheless, the performance of DMLS was still excellent compared to that of the evaluated baseline techniques.

Finally, two key algorithmic optimizations to increase the speed of DMLS were presented. It was shown that these optimizations could be combined to further improve the execution speed of DMLS by a factor of five to six times.

In summary, this paper has demonstrated that DMLS is an excellent candidate for fast unrestricted vocabulary speech document indexing. It offers rapid search speeds while maintaining good detection performance, thus providing clear benefits compared to other approaches such as the conventional lattice search and HMM-based spotting methods.

REFERENCES

[1] J. Makhoul, F. Kubala, T. Leek, D. Liu, L. Nguyen, R. Schwartz, and A. Srivastava, "Speech and language technologies for audio indexing and retrieval," *Proc. IEEE*, vol. 88, no. 8, pp. 1338–1353, Aug. 2000.

[2] A. G. Hauptmann and H. D. Wactlar, "Indexing and search of multimodal information," in *Proc. IEEE Int. Conf.Acoust., Speech, Signal Process.*, Apr. 1997, vol. 1, pp. 195–198.

[3] J. Hansen, R. Huang, B. Zhou, M. Seadle, J. R. Deller, A. R. Gurijala, M. Kurimo, and P. Angkititrakul, "Speechfind:advances in spoken document retrieval for a national gallery of the spoken word," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 5, pp. 712–730, Sep. 2005.

[4] J. R. Rohlicek, *Modern Methods of Speech Processing*. Norwell, MA: Kluwer, 1995, ch. 9, pp. 136–140.

[5] S. Dharanipragada and S. Roukos, "A multistage algorithm for spotting new words in speech," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 8, pp. 542–550, Nov. 2002.

[6] S. J. Young and M. G. Brown, "Acoustic indexing for multimedia retrieval and browsing," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 1997, vol. 1, pp. 199–202.

[7] P. Yu, K. Chen, C. Ma, and F. Seide, "Vocabulary-independent indexing of spontaneous speech," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 5, pp. 635–643, Sep. 2005.

[8] K. Tanaka, Y. Itoh, H. Kojima, and N. Fujimura, "Speech data retrieval system constructed on a universal phonetic code domain," in *Proc. IEEE Workshop Autom. Speech Recognition Understanding,*, Apr. 2001, pp. 323–326.

[9] D. A. James and S. J. Young, "A fast lattice-based approach to vocabulary independent wordspotting," in *IEEE Int. Conf. Acoust., Speech, Signal Process.*, Adelaide, Australia, 1994, vol. 1, pp. 377–380.

[10] K. Thambiratnam and S. Sridharan, "Dynamic match phone-lattice searches for very fast and accurate unrestricted vocabulary keyword spotting," in *IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2005, vol. 1, pp. 465–468.

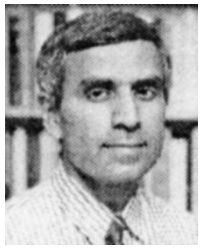
[11] J. G. Wilpon, L. R. Rabiner, C. H. Lee, and E. R. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden Markov models," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 11, pp. 1870–1878, Nov. 1990.

[12] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River, NJ: Prentice-Hall, 2000, ch. 5.

[13] K. Thambiratnam and S. Sridharan, "Isolated word verification using cohort word-level verification," in *Proc. Eurospeech*, Geneva, Switzerland, 2003, pp. 905–908.

Kishan Thambiratnam (M'05) received the B.E. degree in electronics, the B.InfTech. degree in computer science, and the Ph.D. degree in the speech recognition field from the Queensland University of Technology, Brisbane, Australia.

He is currently with the Speech Research Laboratory as a Research Fellow involved in speech indexing and retrieval research. His principal research interests are speech indexing and retrieval, speech confidence scoring, and large-vocabulary speech recognition. Previously, he has been actively involved in related research fields including audio and music identification, indexing and retrieval, automatic speech segmentation, speaker verification, and speech decoding.



Sridha Sridharan (SM'88) received the B.Sc. degree in electrical engineering, the M.Sc. degree in communication engineering from the University of Manchester Institute of Science and Technology (UMIST), Manchester, U.K., and the Ph.D. degree in the signal processing from the University of New South Wales, Sydney, Australia.

He is currently with the Queensland University of Technology (QUT), Brisbane, Australia, where he is a Professor in the School of Electrical and Electronic Systems Engineering. He is the Leader of the Research Program in Speech, Audio, Image, and Video Technologies (SAIVT) at QUT and is a Deputy Director of the Information Security Institute (ISI) at QUT.

Prof. Sridharan is a Fellow of the Institution of Engineers, Australia, and the Chairman of the IEEE Queensland Chapter in Signal Processing and Communication. In 1997, he was the recipient of the award of Outstanding Academic of QUT in the area of research and scholarship.