IDIAP RESEARCH REPORT

Fast Approximate Spoken Term Detection from Sequence of Phonemes

Joel Pinto [a] [b] Igor Szoke [c]
S.R.M. Prasanna [d] Hynek Hermansky [a] [b]

IDIAP–RR 08-45

June 2008

soumis à publication

[a] IDIAP Research Institute, Martigny, Switzerland
[b] École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
[c] Brno University of Technology, Czech Republic
[d] IIT-Guwahati, India

IDIAP Research Institute **www.idiap.ch**
Rue du Simplon 4 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

Rapport de recherche de l'IDIAP 08-45

Fast Approximate Spoken Term Detection from Sequence of Phonemes

Joel Pinto Igor Szoke S.R.M. Prasanna Hynek Hermansky

June 2008

soumis à publication

Résumé. We investigate the detection of spoken terms in conversational speech using phoneme recognition with the objective of achieving smaller index size as well as faster search speed. Speech is processed and indexed as a sequence of one best phoneme sequence. We propose the use of a probabilistic pronunciation model for the search term to compensate for the errors in the recognition of phonemes. This model is derived using the pronunciation of the word and the phoneme confusion matrix. Experiments are performed on the conversational telephone speech database distributed by NIST for the 2006 spoken term detection. We achieve about 1500 times smaller index size and 14 times faster search speed compared to the state-of-the-art system using phoneme lattice at the cost of relatively lower detection performance.

1 Introduction

Spoken term detection (STD) [2] refers to detecting a word or a phrase in unconstrained speech. STD technology is useful in searching large archives of recorded speech such as conversational telephone speech, multi-party meetings etc.

The system consists of two stages - indexing and searching. In the indexing stage, speech is processed and stored in an easy to search form known as index. This involves tagging speech using the sequence of recognized words or phonemes. In the search stage, the index is searched and the exact location of the term is determined.

Spoken term detection can be approached in two ways that differ in the basic unit used for indexing. In the word lattice based approach, the index is a word lattice and is obtained by large vocabulary continuous speech recognition (LVCSR). In the phoneme lattice based approach, the index is a phoneme lattice and is obtained by phoneme recognition.

The current dominant approaches to STD are based on the use of word lattices obtained from LVCSR. A word lattice is a compact representation of the multiple word hypotheses for a given speech utterance. The posterior probability of a word conditioned on the entire utterance can be computed from the word lattice using the forward backward re-estimation algorithm. The word posterior probability was first proposed as a confidence measure in LVCSR [12] [4]. Subsequently, it has been successfully applied for spoken term detection in the 2006 NIST evaluation [7] [11] [10].

The major drawback in the word lattice based approach is its inability to detect out-of-vocabulary words in the LVCSR system. It is often the case that search terms of interest are named entities (names of persons, places, etc) which evolve over time and updating the language model and dictionary is necessary [3]. Moreover, recognition errors may also be introduced as the ASR language model forces meaningful sentences [3].

The above drawbacks can be effectively addressed by using an vocabulary independent approach for indexing such as phoneme recognition. However, the accuracy of recognition of phonemes in conversational speech is only about 50-60% because useful prior knowledge such as word language model and pronunciation dictionary is not used. This problem may be reduced to a certain extent by using a phoneme lattice instead of a one best sequence of phonemes.

A phoneme lattice represents multiple phoneme hypotheses for a given speech utterance. Unlike the word lattice, the phoneme lattice does not contain any words. Hence, in the search stage, the pronunciation of the required search term is searched in the phoneme lattice using dynamic time alignment based methods [6] [8].

Apart from the performance, the processing requirements such as the index size, indexing time, and the search time determine the feasibility of a practical large scale deployment of the STD system. Word lattice based STD systems require a large index space and indexing is slow, but search is instantaneous. On the other hand, in phoneme lattice based STD system, the size of the index remains more or less the same, indexing is relatively faster, but search is slower.

With an objective of achieving smaller index size as well as faster search speed in a phoneme based STD, we investigate the use of one best phoneme sequence for indexing and a probabilistic pronunciation model for the search term. The pronunciation model uses information from the phoneme confusion matrix to compensate for the errors in the recognition of phonemes. We compare our approach to the state-of-the art STD system using phoneme lattices [9]. It is expected that the performance of the proposed system will be inferior compared to the lattice based system. However, the proposed method will result in significantly smaller index size and faster search speed.

The potential applications of the proposed approach include (a) searching very large archives such as speech/multimedia content available on the web, in which case the storage space for the index is paramount, and (b) fast preselection of speech utterances to be subsequently searched using more sophisticated methods.

The rest of the paper is organized as follows : Section 2 explains the background to our approach to STD, Section 3 describes the probabilistic pronunciation model for the search term and its detection. The phoneme lattice based state-of-the-art system used for comparison is discussed in Section 4.

The experimental setup is described in Section 5 and results are presented in Section 6. Finally, the summary and future scope is discussed in Section 7.

2 Background

In STD, the search terms are specified as a sequence of words, but the basic acoustic modeling units are typically phonemes. Hence, there is a need for a phonetic pronunciation dictionary, which maps the words to its correct pronunciation. Dictionaries also include multiple pronunciations for a word to account for the known pronunciation variants. In this work, we use a probabilistic model for the pronunciation of the search terms to compensate for the errors in the one best recognition of phonemes.

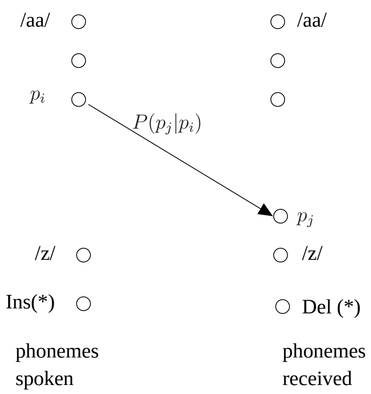

Fig. 1 – Noisy channel model for the phoneme recognizer

Errors in the recognition of phonemes could be due to (i) mispronunciation by the speaker, or (ii) inaccurate modeling of phonemes leading to acoustic confusion. To capture the systematic errors in the recognition of phonemes due to acoustic confusion, we treat the phoneme recognition system as a discrete, memoryless, and noisy channel. As shown in Figure 1, the phonemes spoken are taken as the source symbols and the recognized phonemes as received symbols. Moreover, an additional symbol '*' at the input denotes insertion and the symbol '*' at the output denotes the deletion of a phoneme. The properties of such a simplified model can be captured using the following parameters.

- $P_i(p_i)$: Insertion probability of phoneme p_i and can be interpreted as $P(p_i|*)$ in Figure 1.
- $P_s(p_j|p_i)$: Probability of phoneme p_j substituted for p_i. The case where $p_j = *$ denotes the deletion of p_i.
- P_s : Unconditional probability of substitution. Deletion is a special case of substitution.
- P_i : Unconditional probability of insertion, $P_s + P_i = 1$.

The above probabilities are estimated by dynamic time alignment of the reference phoneme sequence to the recognized phoneme sequence and normalizing the substitution, insertion, and deletion counts. The estimated parameters of a noisy channel model for the phoneme recognizer captures its error characteristics and is generally referred to as the confusion matrix.

In modeling the search term, the confusion matrix obtained from a noisy channel model is treated as the prior knowledge of the phoneme recognizer. The probabilistic pronunciation model for the search term is derived using the correct pronunciation, obtained from the dictionary and the noisy channel model of the phoneme recognizer as explained in the following section.

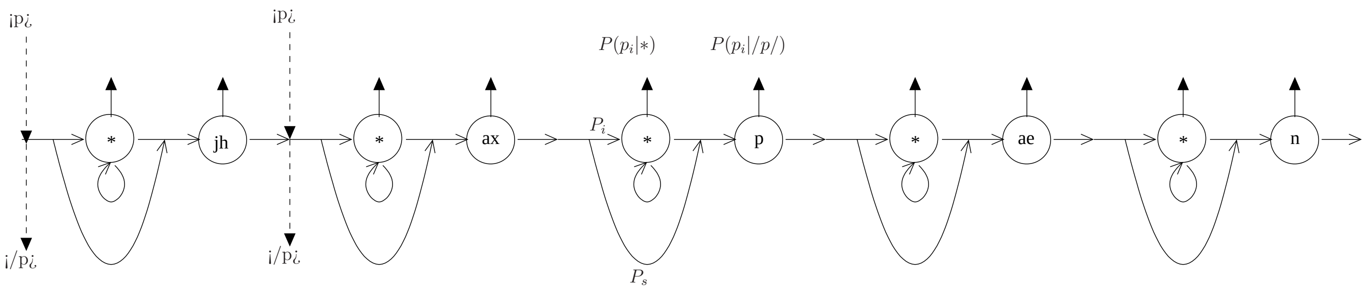

Fig. 2 – A probabilistic pronuciation model for a search term 'JAPAN' with a correct pronuciation of /jh/ /ax/ /p/ /ae/ /n/. Symbol '*' represents the hidden state for insertion. ¡p¿ and ¡/p¿ denotes the entry to and exit from the model (not shown for every phoneme).

3 STD using Phoneme Sequence

A search term is modeled using an hidden Markov model (HMM). The phonemes in the correct pronunciation for the search term are taken as the hidden states of the model. Moreover, there is an optional hidden state ('*') for insertion that can be entered with certain probability P_i. The emission probability in the insertion state is given by the phoneme insertion probability from the confusion matrix. The emission probability in the phonemic state p_i is given by the conditional probability $P(p_j|p_i)$ from the confusion matrix. The special case where the phoneme p_j = ' $*$ ' represents the deletion of phoneme p_j. The topology of the model is determined by the correct pronunciation of the search term. The model parameters are obtained from the phonetic confusion matrix.

Figure 2 is a schematic of the proposed pronunciation model for the word 'Japan' with pronunciation /jh/ /ax/ /p/ /ae/ /n/. This model can be used to generate the sequence of phonemes likely to be observed at the output of the recognizer when the word 'Japan' is spoken. We further relax the model by letting it begin and end at any phoneme in pronunciation. The symbol '¡p¿' denotes the entry to the model and '¡/p¿' denotes exit from the model. By fixing a parameter N, the relaxed model can generate all N-length sequence of phonemes that are likely to be observed at the output of the recognizer when the word 'Japan' is spoken. Conversely, given any N-length sequence of observed phonemes, we could find the state sequence in the model that is most likely to have generated the observed phoneme subsequence. Subsequently, the conditional probability $P(\textit{observed seq}|\textit{state seq}, \textit{word model})$ is taken as the score for the observed phoneme sequence given the model. The three cases of substitution, insertion, and deletion are described below with example for $N = 3$.

Substitution :

observed phoneme sequence (q) = {jh, ey, p}
aligned observation sequence($q\prime$) = {jh, ey, p}
aligned best state sequence (s) = {jh, ax, p}

$$P(q\prime|s, W) = P(jh|jh)\ .\ P(ey|ax)\ .\ P(p|p)$$

Deletion :

observed phoneme sequence (q) = {jh, ax, ae}
aligned observation sequence($q\prime$) = {jh, ax, *, ae}
aligned best state sequence (s) = {jh, ax, p, ae}

$$P(q\prime|s, W) = P(jh|jh)\ .\ P(ax|ax)\ .\ P(*|p)\ .\ P(ae|ae)$$

Insertion :

observed phoneme sequence (q) = {jh, ax, ax}
aligned observation sequence$(q\prime)$ = {jh, ax, ax}
aligned best state sequence (s) = {jh, *, ax }

$$P(q\prime|s, W) = P(jh|jh)\ .\ P(ax|*)\ .\ P(ax|ax)$$

This proposed approach can be seen as an extension of the work in [5], where a similar model was investigated for keyword spotting. Our work differs from the previous study in the following aspects : (a) precise formulation of the problem as a probabilistic pronunciation model (b) best state sequence is obtained by dynamic time alignment, (c) usage of the estimated conditional probabilities in word detection, and (d) using this method on conversational speech as opposed to the simulated sequence of phonemes.

To search for a particular term in sequence of phonemes, we first apply a sliding window N phonemes long with a shift of 1 phoneme. The probability of the windowed phoneme subsequence given the model for the search term is computed. The estimated probability is expected to be high when the search term is indeed present and to be low otherwise. Figure 3 shows the log-probability of the phonemes in the sliding window given the model for the word 'Arizona'.

One way of detecting the word would be to simply apply a preselected threshold on the log-probabilities. In this work, we use a garbage model to estimate a reference score and apply Viterbi decoding. Moreover, since the search term and hence the number of phonemes is known a priori, we exploit this by enforcing minimum duration constraints while detection.

3.1 Garbage Model

In keyword spotting, garbage models are used to absorb any non-keyword speech. Typical garbage modeling techniques include explicit training of an HMM or GMM on non-keyword speech, an ergodic network of trained phoneme models etc. The garbage model provides a reference score to compare the keyword model score. In this work we propose the use of a N-gram phoneme language model as a garbage model.

N-gram phoneme language models are typically used to capture phonotactic prior knowledge in phoneme recognition. N-gram model estimates the probability $P(q_k|q_{k-N+1}^{k-1}, G)$, where q_{k-N+1}^{k-1} denotes the $N-1$ phonemes proceeding q_k. We use the joint probability $P(q_{k-N+1}^{k}|G)$, which can be obtained using Baye's rule as the garbage model score.

Figure 3 is a plot of the log-probability of the phoneme in the sliding window given the model for the word 'Arizona' as well as the garbage model. It can be seen that when the word 'Arizona' is present in the phoneme sequence, the word model score is higher than the garbage model score.

The search term and the garbage models are modeled using a left-right HMM with certain number of states (M_s), which is proportional to the number of phonemes (M_p) in the search term and is given by $M_s = M_p - N + 1$. In the case of longer search terms, the minimum duration could be further reduced to increase detection rate. The emission probability in every state of the search term is $P(q|s, W)$, obtained from the proposed method pronunciation model. The emission probability in each of the garbage model state is taken as $P(q|G)$. Viterbi algorithm is applied to detect the presence of the search term. By using minimum duration constraint, the spurious peaks that are likely to occur due to the relaxed pronunciation model are suppressed.

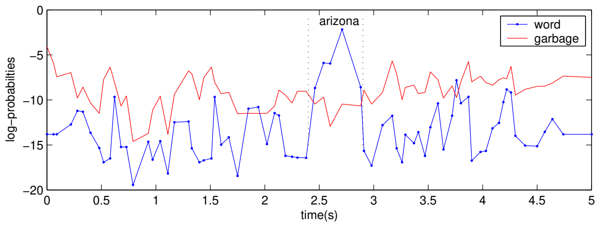

FIG. 3 – Log-probability of the phonemes in the sliding window along the observed phoneme sequence, given the pronunciation model for the word 'Arizona', and the garbage model.

4 STD using Phoneme Lattice

We compare the proposed method to the state-of-the-art system where the pronunciation for the search term is searched in the phoneme lattice [8]. The phoneme lattice is a directed acyclic graph whose nodes represents the time information and the arcs contains the phoneme and its acoustic model likelihood. Phoneme language model probabilities are not considered in the search.

The confidence measure $C(S)$ for the hypothesized search term S with begin time t_b and end time t_e is given by

$$C(S) = L_\alpha(t_b) + l(S) + L_\beta(t_e) - L_{best}, \tag{1}$$

where, $L_\alpha(t_b)$ denotes the accumulated forward log-likelihood from the start of the lattice to the search term S, $L_\beta(t_e)$ is the accumulated backward log-likelihood from the end of the lattice to end of the search term, L_{best} denotes the log-likelihood of the best path through the lattice, and $l(S)$ denotes the log-likelihood of the search term and is given by

$$l(S) = \sum_{i=1}^{K} [\phi_i \; l_a(i) + (1 - \phi_i) \; l_w]. \tag{2}$$

The variable $\phi_i = 1$ if the i^{th} phoneme in the search term matches with the i^{th} phoneme in the K-length substring of the lattice. $l_a(i)$ is the acoustic likelihood for the i^{th} phoneme, and l_w is the penalty for the mismatch.

The accumulated forward and backward log-likelihoods at a node N in the phoneme lattice are recursively computed as

$$L_\alpha(N) = \max_{N_P} \; [l_a(N_P, N) + L_\alpha(N_P)] \tag{3}$$

$$L_\beta(N) = \max_{N_F} \; [l_a(N, N_F) + L_\beta(N_F)] \tag{4}$$

where, N_P is the set of nodes preceding N in the phoneme lattice and $l_a(N_P, N)$ is the corresponding acoustic model log-likelihood. Similarly, N_F is the set of nodes following N and $l_a(N_P, N)$ is the log-likelihood for the arc. The first node has $L_\alpha(first) = 1$, and the last node has $L_\beta(last) = 1$.

By Viterbi search, we obtain several overlapping hypothesis for the search term with different begin and end times and confidence measure. We select the hypothesis with the highest confidence measure. The final hypothesis could be either accepted or rejected by comparing the confidence measure to a pre-selected threshold.

5 Experimental Setup

Experiments were performed on three hours of two-channel conversational telephone speech (CTS) development database distributed by NIST for the 2006 spoken term detection evaluation (STD06) [2].

A neural network based speech-silence algorithm is applied to obtain 5257 utterances comprising three hours of speech to be searched.

The search terms were selected from the dryrun set distributed for the STD06 evaluation. Of the 1107 search terms in the list, we select only those terms that occur at least once in the test speech. We also drop the search terms with fewer than four phonemes as any phoneme based STD will give high false alarms [3]. The final search list consists of 153 single-word, 78 two-word, and 12 three-word search terms.

5.1 Acoustic Modeling

The acoustic models are based on the hidden Markov model - Gaussian mixture models, trained using multi-pass training strategy. These models were trained using 277 hrs of *ctstrain04* database, which is a subset of *h5train03* set defined at Cambridge university.

The base features comprises of 12 Mel frequency perceptual linear predictive coefficients and raw log-energy. Feature normalization techniques such as cepstral mean/variance normalization, vocal tract length normalization are applied. Model adaptation techniques such as maximum likelihood linear regression (MLLR) and constrained MLLR are also used. These features are finally appended to the posterior based features from a neural network. Feature extraction and training of the acoustic models was done at Brno University of Technology. A detailed description of the system can be found at [10].

5.2 Phoneme Recognition

Context-dependent acoustic models were used for recognition of phonemes with a bigram phoneme language model. Posterior based pruning was applied to obtain compact lattices. These phoneme lattices were subsequently used for searching in the baseline method.

Viterbi decoding was applied on the phoneme lattice to obtain one best sequence of phonemes to be used in the proposed method. We obtain a phoneme recognition accuracy of 64%.

5.3 Evaluation Metric

The performance of the proposed method is evaluated using the actual term-weighted value (ATWV) [1] defined by NIST for STD06 evaluation, and is defined as

$$ATWV = 1 - \frac{1}{T}\sum_{t=1}^{T}(P_{Miss}(t) + \beta P_{FA}(t)),$$

where, the probability of miss ($P_{Miss}(t)$) and probability of false alarm $P_{FA}(t)$ for the term t is given by

$$P_{Miss}(t) = 1 - \frac{N_{corr}(t)}{N_{true}(t)}, \;\; and \;\; P_{FA}(t) = \frac{N_{spurious}(t)}{T_{speech} - N_{true}(t)}.$$

$N_{corr}(t)$ is the number of correct detections, $N_{true}(t)$ is the number of true occurrences of the term in the test corpus, $N_{spurious}(t)$ denotes the number of spurious detections of the term, and T_{speech} is the duration of the test speech in seconds. β is set to be 1000.

A perfect system ($P_{Miss} = 0, P_{FA} = 0$) has an $ATWV = 1.0$. A system that does nothing ($P_{Miss} = 1, P_{FA} = 0$) has an $ATWV = 0$. Assuming that $T_{speech} >> N_{true}$, if the system gives perfect detection ($P_{Miss} = 0$), but 3.6 false alarms per hour for each term, we get an $ATWV = 0$. This shows that ATWV is a stricter evaluation metric compared to figure-of-merit which is typically used in evaluating keyword spotting systems.

6 Results

To study the performance of the proposed STD system for different lengths (in term of number of phonemes) of the search term, the search terms are sub-divided into groups containing 5-6, 7-8, 9-10, 11-13, 14-16, and above 17 phonemes. The pronunciation for search terms with multiple words is obtained by concatenating the individual word pronunciations.

Tab. 1 – ATWV of the proposed method compared to the baseline system.

phonemes	5-6	7-8	9-10	11-13	14-16	17+
baseline	0.55	0.69	0.80	0.73	0.74	0.71
proposed(N=3)	0.23	0.27	0.45	0.60	0.70	0.94

Table 1 shows the ATWV for the proposed probabilistic pronunciation model as compared to the state-of-the-art phoneme lattice based system. It can be seen that the phoneme lattice based approach outperforms the proposed method, especially for search terms with fewer number of phonemes. This is because the confusions captured by the phoneme lattice $P(p_j|p_i, X)$ is conditioned on the observed data X and hence is more accurate. On the other hand, the confusion matrix $P(p_j|p_i)$ used in the proposed approach is averaged over all the data.

6.1 Processing Requirements

In spoken term detection, emphasis is also on (i) size of the index, (ii) indexing speed, and (iii) search speed. Table 2 shows the index size, indexing speed and the average search speed for the proposed method as compared to the baseline phoneme lattice based approach.

Tab. 2 – Indexing size, indexing speed, and search speed for the proposed system as compared to the phoneme lattice based system. Notations are MB :megabytes, Hs :hours of speech, Hp :processing hours, and sp :processing seconds.

	index size (MB/Hs)	indexing speed (Hp/Hs)	search speed (sp/Hs)
baseline	187.3	86.0	13.5
proposed	0.127	86.0	1.0

It can be seen from the Table 2 that the proposed method needs about 1500 times less memory for storing the index and is about 14 times faster when compared to the phoneme lattice based system. Moreover, during search, building the probabilistic pronunciation model is takes most of the computational time. Once the pronunciation model is estimated, searching takes 0.52 sp/Hs which is approximately 7000 times faster than real time. Thus, at the expense of reduced performance, we achieve an order of three magnitude reduction in the size of the index and order of magnitude increase in search speed.

6.2 Effect of phoneme recognition accuracy

Since the proposed method uses a one best sequence of phonemes for detecting the search terms, we analyze the performance of the system for different phoneme recognition accuracies. Different recognition accuracies are simulated by replacing the phonemes in the reference transcription by randomly generated phonemes. The probability mass function for the random generation is obtained from the confusion matrix. By controlling the number of substitutions, we simulate recognition accuracies between 100% and 65%.

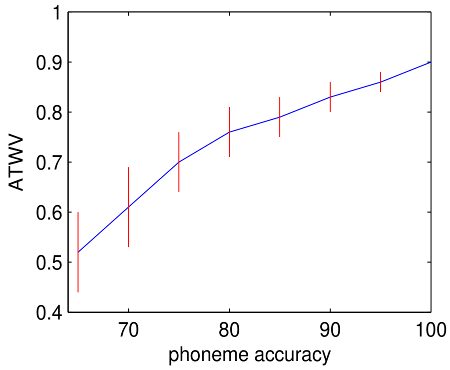

Fig. 4 – Plot of the mean ATWV (and the standard deviation) measure for simulated phoneme recognition accuracies.

Figure 4 shows the performance of the STD for search terms containing 11-13 phonemes for different accuracies of recognition of phonemes. It can be seen that perfect phoneme recognition does not give an ATWV of 1.0. Analysis of the results reveal that this is mainly because of the pronunciation ambiguities that cannot be resolved in a phoneme based methods. Inaccurate speech silence segmentation leading to a multi-word spoken term segmented into different utterances is also observed. It can also be seen that improved phoneme recognition greatly improves the ATWV. This justifies the renewed research interest in recognition of phonemes.

7 Discussion And Future Work

Traditionally, in keyword spotting as well as in spoken term detection using phoneme recognition, the pronunciation of the keyword is searched in the phoneme lattice using dynamic time alignment based methods. In this work, we use a one best sequence of phonemes and investigated a probabilistic model for the keyword. The pronunciation model is derived from the word pronunciation and the phonetic confusion matrix. Both the phoneme lattice as well as the confusion matrix capture the confusion among phonemes. The phonetic confusion captured by the phoneme lattice is conditioned on the observed data and is more accurate compared to the confusion matrix. This is also reflected in the experimental results.

In this work, we present the preliminary results obtained using a probabilistic pronunciation model. We have used a very simplistic model to estimate the phonetic confusions in a phoneme recognizer. Future work include the use of more informative models such as bigram confusion matrix $P(p_i, p_j|p_k, p_l)$.

8 Conclusions

We demonstrate a fast approximate spoken term detection system that uses one best sequence of recognized phonemes as the index. A probabilistic pronunciation model for the search term is used to compensate the errors in recognition of phonemes. As expected, the performance of such system is inferior compared to the state-of-the-art phoneme lattice based system. However, the proposed approach could be particularly useful in practical scenarios where the size of the index, and the search speed are critical.

9 Acknowledgements

This work was supported in parts by the Swiss National Science Foundation under the Indo-Swiss joint research program KEYSPOT, the European Union under the DIRAC integrated project, contract No. FP6-IST-027787 as well as DARPA under the GALE program, contract No. HR0011-06-C-0023. Any findings and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of funding agencies.

Références

[1] NIST Spoken Term Detection Evaluation Plan, 2006. *http ://www.nist.gov/speech/tests/std/2006/docs/std06-evalplan-v10.pdf*.

[2] NIST Spoken Term Detection Evaluation. *http ://www.nist.gov/speech/tests/std*, 2006.

[3] P. Cardillo, M. Clements, and M. Miller. Phonetic searching vs Large vocabulary continuous speech recognition. *International Journal of Speech Technology*, 2002.

[4] G. Evermann and P. Woodland. Large Vocabulary Decoding and Confidence Estimation using Word Posterior Probabilities. *Proc. of IEEE Conf. Acoust. Speech. Signal Process. (ICASSP)*, 2000.

[5] A. Ito and S. Makino. A New Word Pre-selection Method based on an Extended Redundant Hashaddressing for Continuous Speech Recognition. *Proc. of IEEE Conf. Acoust. Speech. Signal Process. (ICASSP)*, 1993.

[6] D. James and S. Young. A Fast Lattice-Based Approach to Vocabulary Independent Wordspotting. *Proc. of IEEE Conf. Acoust. Speech. Signal Process. (ICASSP)*, 1994.

[7] D. Miller et al. Rapid and Accurate Spoken Term Detection. *In Proc. of NIST Spoken Term Detection Workshop (STD 2006)*, Dec 2006.

[8] I. Szoke et al. Comparison of Keyword Spotting Approaches for Informal Continuous Speech. *Proc. of Interspeech*, 2005.

[9] I. Szoke et al. BUT System for NIST Spoken Term Detection 2006 - English. *In Proc. of NIST Spoken Term Detection Workshop (STD 2006)*, 2006.

[10] I. Szoke et al. Combination of Word and Phoneme Approach for Spoken Term Detection. *4th Joint Workshop on Machine Learning and Multimodal Interaction*, 2007.

[11] D. Vergyri et al. The SRI/OGI 2006 Spoken Term Detection System. *Proc. of Interspeech*, 2007.

[12] F. Wessel, R. Schlute, K. Macherey, and H. Ney. Confidence Measures in Large Vocabulary Speech Recognition. *IEEE Transactions on Speech and Audio Processing*, 9(3), March 2001.