A Hybrid Keyword Spotting Approach for Combining LVCSR and Phonetic Search

Ella Titariy, Noam Lotner, Michal Gishri, Ami Moyal

The Afeka Center for Language Processing (ACLP)
Afeka Academic College of Engineering, Tel Aviv, Israel

Abstract - Speech indexing and retrieval has become more and more crucial with the constant accumulation of massive amounts of digital data. When it comes to audio and video data, speech recognition technology is often used in Keyword Spotting (KWS) based applications to enable specific words to be identified out of a stream of continuous speech.

It is commonly recognized that current KWS solutions provide acceptable results for well-resourced languages, but that they fall short for under-resourced languages. This paper will describe a method for performing keyword spotting in under-resourced languages by merging the results of both LVCSR-based and Phonetic Search based keyword spotting engines. The KWS results from each of the engines are optimized by applying thresholds to each keyword.

Experiments were conducted on English, Vietnamese and Tamil. The results show that the merging mechanism used was able to improve the Term-Weighted Value (TWV) [1] score by 6-12% over the best single-engine result when conducted on development data, but reduced the TWV score when performed using evaluation data. Possible reasons for this degradation in performance are discussed.

1 INTRODUCTION

For the experiments presented in this paper, keyword spotting was conducted on each of the three languages using four separate engines. 1) a phonetic search based engine that compares the phoneme sequences of keywords with the phoneme sequences produced by a phoneme decoder; 2) an LVCSR-based engine in which a textual keyword search is performed over the one-best transcription of the speech; 3) a combined LVCSR-phonetic search engine that performs phonetic search over a phoneme sequence produced by transforming the one-best LVCSR results to phonemes; and 4) an LVCSR-based engine in which a textual keyword search is performed over a word lattice produced by the LVCSR engine.

Keyword-dependent thresholds were assigned in order to optimize performance of each of the four base engines. The base engine results were then fused using a best-engine merging mechanism which assigns the best performing engine to each separate keyword.

2 KWS BASE ENGINES

2.1 LVCSR

Performing KWS on databases transcribed by a Large Vocabulary Continuous Speech Recognition (LVCSR) engine is relatively straightforward. First, an LVCSR engine is employed to transcribe the entire speech signal into text. The LVCSR engine produces the most probable sequence of words based on the Viterbi search algorithm, using acoustic models, a large lexicon of words and a language model. In the second stage, the KWS mechanism performs a search on the resulting speech-aligned text to locate the keywords.

The two LVCSR-based engines that were employed are as follows:

1) **LVCSR 1-best:** This engine performs the keyword search on the one-best LVCSR transcription results.
2) **LVCSR-Lattice:** This engine performs the keyword search on a lattice of words.

2.2 Phonetic Search

Phonetic search KWS is implemented in two sequential stages. The first stage converts the speech database into sequences of phonemes. This initial transformation of speech into phonemes is a one-time, fast, and usually off-line, procedure. In the second stage, the actual keyword search is performed by matching the phoneme sequences representing the keywords to phoneme sequences found in the database (keyword hypotheses), and selecting exact or near-matching sequences as outputs for a list of detected keywords. The matching process is performed using a weighted Levenshtein distance [2, 3] that incorporates variable penalties based on statistics extracted from the decoding results of the development data.

Two phonetic search-based engines that were employed are as follows:

1) **PD Sequence:** This engine performs a phonetic search on the phonetic sequences produced when the speech is run through a Phoneme Decoder.
2) **LVCSR-Conversion:** This engine converts the one-best LVCSR transcription to phoneme sequences (using the original pronunciation lexicon used by the LVCSR engine), and performs a phonetic search on these sequences.

3 KEYWORD-DEPENDENT THRESHOLDING

A basic system uses a uniform adjustable threshold for deciding which candidates are accepted as valid detection hypotheses. However, experience shows that using a different threshold value for each keyword improves the overall performance of a system both in terms of increasing keyword detections and in reducing the False Alarm Rate (FAR).

Using the development data, and comparing detection results with the reference, the threshold that produced the best TWV score for each keyword was calculated separately for each of the four engines. The threshold scale used for the 1-best and lattice based engines was the confidence score produced by the LVCSR engine. The threshold scale used for the phonetic search engine and the LVCSR word-to-phoneme conversion engine was the weighted Levenshtein Distance measure produced by the phonetic search engine.

4 EXPERIMENTAL FRAMEWORK

4.1 Language Resources

The experiments on English were conducted using the NIST 2006 Spoken Term Detection Database. The English acoustic models were trained using a collection of recordings from the Fisher 1 [4], MACROPHONE [5] and CSLU ver. 1.1 [6] databases. The experiments on Vietnamese were conducted on the NIST 2013 Open Keyword Search surprise language training, development and evaluation data and on Tamil using the NIST 2014 Open Keyword Search training and development data (evaluation results are not presented in this paper). Table 1 summarizes the data used in the experiments.

	English	Vietnamese	Tamil
Training	~800 hrs	80 hrs	66[1] hrs
Development	3 hrs	10 hrs	10 hrs
Dev. Keywords	1099	200	2000
Evaluation	6 hrs	77 hrs	75 hrs
Eval. Keywords	1100	4065	5576

Table 1 Language Resources Utilized in Experiments

4.2 Term Weighted Value Scoring Metric

In order to provide a basis for comparison between the experimental results, it is necessary to use a single performance metric. The metric selected was the Term Weighted Value (TWV) used for the Open Keyword Search 2014 (OpenKWS14) Evaluation [7] and is defined as follows:

The TWV is 1 minus the weighted sum of the term-weighted probability of missed detection (PMiss(θ)) and the term-weighted probability of false alarms (PFA(θ)).

$$TWV(\theta) = 1 - [P_{Miss}(\theta) + \beta \cdot PFA(\theta)]$$

The Actual TWV (ATWV) is the actual working point of the system and the Maximum TWV (MTWV) is the maximum value on the system DET curve [1].

4.3 Word and Phoneme Error Rates

Prior to running the KWS experiments, the word and phoneme error rates were calculated for each of the three languages in order to get an indication of the quality of the acoustic models and the LVCSR and phoneme decoder output. All acoustic models were trained using the KALDI open source speech recognition toolkit [8]. Table 2 shows the Phoneme Error Rate (PER) and Word Error Rate (WER) obtained for each language on its respective development databases.

	PER	WER
English	45.96%	40.49%
Vietnamese	54.49%	54.63%
Tamil	55.71%	68.19%

Table 2 PER and WER for English, Vietnamese and Tamil

5 BASE SYSTEM RESULTS

Error! Reference source not found. shows the performance results of each of the 4 individual KWS engines on the development data for all three languages using the development keyword list (see Table 1). The Maximum TWV thresholding mechanism was used to produce all performance results. Dynamic anchoring for reducing computation complexity [9, 10] was used on the phonetic search procedures (PD Sequence and LVCSR-Conversion systems).

The performance on the evaluation data is lower in most cases, however this is to be expected, as the keyword-dependent thresholds were tuned on the development database, giving the KWS results somewhat of a bias when the system is both tuned and tested on the same database. **Error! Reference source not found.** presents the same results on the evaluation databases. Note that at the time this paper was presented, the final evaluation results on Tamil could not be presented due to the OpenKWS14 confidentiality conditions. Thus the evaluation results shown are for English and Vietnamese only.

6 BEST ENGINE MERGING MECHANISM

All base engine experiments show that the LVCSR-Lattice engine consistently out-performs the other three engines. The goal of any merging mechanism would

[1] The 20 hours of untranscribed data provided with the OpenKWS14 Build Pack were not utilized in the training process.

MTWV Scores	**LVCSR**		**Phonetic Search**	
	1-Best	**Lattice**	**PD Sequence**	**LVCSR Conversion**
English	0.5877	0.7424	0.2724	0.5304
Vietnamese	0.2173	0.3755	0.1024	0.2831
Tamil	0.2086	0.4097	0.1165	0.2783

Table 3 MTWV Scores for Development Keywords tested on Development Databases

MTWV Scores	**LVCSR**		**Phonetic Search**	
	1-Best	**Lattice**	**PD Sequence**	**LVCSR Conversion**
English	0.5190	0.5731	0.0734	0.3026
Vietnamese	0.1789	0.2989	-0.2431	-0.0943

Table 4 MTWV Scores for Evaluation Keywords tested on Evaluation Databases

therefore be to reach a TWV score that is higher than the result produced by the LVCSR Lattice engine alone. The notion that this is possible lies in the assumption that not all of the system KWS decisions are intersecting, and specifically, that an LVCSR engine alone cannot produce results for OOV keywords. If this assumption is correct, even a system that produces relatively poor KWS results on its own may have a positive effect in a merge by either contributing to detections or obstructing false alarms.

The 'best engine' merging mechanism analyzes which of the four engines outputs the best results (highest TWV) for each individual keyword after applying thresholds. Both the development and the evaluation keywords were tuned on the development databases to define the best engine for each of the keywords.

OOV keywords, which are not part of the pronunciation lexicon, and therefore cannot be found in the LVCSR output transcriptions, were *not* handled separately, but were assigned a Maximum TWV (MTWV)-based threshold using the same procedure. In this case, however, the threshold leading to the MTWV will essentially block false alarms. The same is true for evaluation keywords that are Out-of-Reference (OOR). These words are not found in the development database making it impossible to estimate thresholds based on the referenced data.

MTWV Scores	**Best Engine Merge**	
	Development	**Evaluation**
English	0.7953	0.5739
Vietnamese	0.4287	0.2899
Tamil	0.4452	

Table 5 Comparison of MTWV on Development vs. Evaluation Databases

Table 5 shows the MTWV score resulting from the merge on both the development and evaluation databases.

It is clear that on the development data the merging mechanism works well, showing a 6-12% improvement in the MTWV score from the LVCSR lattice engine, depending on the language. Using the evaluation database English and Vietnamese show degradation in performance between the LVCSR Lattice results and the Best Engine merging mechanism.

7 DISCUSSION AND FUTURE RESEARCH

Based on the results presented in this report, there are several steps that should be taken to improve the KWS approach used. While the direction of merging the results from several engines is promising, the actual merging mechanism needs to be refined to include score normalization, and weighing of base engine results rather than a simple best engine merge. Furthermore, it is necessary to develop a more robust thresholding mechanism which handles both OOV and OOR keywords separately.

Additional future research directions also include increasing the amount training data by including available untranscribed data and testing all mechanisms on several languages and utilizing a DNN (Deep Neural Network) as acoustical models.

8 REFERENCES

1. *2014 Open Keyword Evaluation Plan, v-11*. 2013, The National Institute of Standards and Technology (NIST).
2. Pucher, M., et al. *Phonetic Distance Measures for Speech Recognition Vocabulary and Grammar Optimization*. in *3rd Congress of the Alps Adria Acoustics Association*. 2007. Graz, Austria.

3. Hermelin, D., et al. *A Unified Algorithm for Accelerating Edit-Distance Computation via Text Compression*. in *26th International Symposium on Theoretical Aspects of Computer Science*. 2009. Feiburg.
4. Christopher Cieri, D.G., Owen Kimball, Dave Miller, Kevin Walker, *Fisher English Training Speech Part 1 Speech LDC2004S13*. 2004, Linguistic Data Consortium: Philadelphia.
5. Jared Bernstein, K.T., Jack Godfrey, *MACROPHONE LDC94S21*. 1994, Linguistic Data Consortium: Philadelphia.
6. CSLU, *CSLU: Speaker Recognition Version 1.1 LDC2006S26*. 2006, Linguistic Data Consortium: Philadelphia.
7. *2014 Open Keyword Evaluation*. 2014 [cited March, 2014]; Available from: http://nist.gov/itl/iad/mig/openkws14.cfm.
8. Povey, D.a.G., Arnab and Boulianne, Gilles and Burget, Lukas and Glembek, Ondrej and Goel, Nagendra and Hannemann, Mirko and Motlicek, Petr and Qian, Yanmin and Schwarz, Petr and Silovsky, Jan and Stemmer, Georg and Vesely, Karel. *The Kaldi Speech Recognition Toolkit*. in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. 2011. Hawaii: IEEE Signal Processing Society.
9. Tetariy, E., V. Aharonson, and A. Moyal. *Phonetic Search Using an Anchor-Based Algorithm*. in *Proceedings of IEEE 26th Convention of Electrical and Electronics Engineering in Israel*. 2010. Eilat.
10. Tetariy, E., et al., *An efficient lattice-based phonetic search method for accelerating keyword spotting in large speech databases.* International Journal of Speech Technology, 2012.