Graph-based Re-ranking using Acoustic Feature Similarity between Search Results for Spoken Term Detection on Low-resource Languages

Hung-yi Lee, Yu Zhang, Ekapol Chuangsuwanich, James Glass

MIT Computer Science and Artificial Intelligence Laboratory
Cambridge, Massachusetts 02139, USA
{tlkagk, yzhang87, ekapolc, glass}@csail.mit.edu

Abstract

Acoustic feature similarity between search results has been shown to be very helpful for the task of spoken term detection (STD). A graph-based re-ranking approach for STD has been proposed based on the concept that search results, which are acoustically similar to other results with higher confidence scores, should have higher scores themselves. In this approach, the similarity between all search results of a given term are considered as a graph, and the confidence scores of the search results propagate through this graph. Since this approach can improve STD results without any additional labelled data, it is especially suitable for STD on languages with limited amounts of annotated data. However, its performance has not been widely studied on benchmark corpora. In this paper, we investigate the effectiveness of the graph-based re-ranking approach on limited language data from the IARPA Babel program. Experiments on the low-resource languages, Assamese, Bengali and Lao, show that graph-based re-ranking improves STD systems using fuzzy matching, and lattices based on different kinds of units including words, subwords, and hybrids.
Index Terms: Random Walk, Spoken Term Detection

1. Introduction

This paper focuses on spoken term detection (STD) [1, 2], in which the query is a keyword[1] in text form, and the goal is to return the time span of all occurrences of the keyword in a spoken archive. In word-based STD, two processing stages are used [3, 4]. The audio content is first transcribed into lattices, and when a user enters a keyword, the STD system searches through the lattices, and returns a list of time spans hypothesized to be the keyword. Based on this approach, the retrieval performance is highly dependent on the ASR output quality, so it works well when the recognition accuracy is high, but becomes less adequate for retrieving spoken archives produced in languages without sufficient resources for training a high quality ASR system [5].

Because different instances of a given keyword will have similar pronunciations, and thus similar acoustic feature sequences, an STD system can augment ASR-based retrieval performance by exploiting acoustic feature similarity between the time spans hypothesized to be the same keyword [6, 7, 8]. Graph-based re-ranking is one way to realize this idea [9, 10, 11, 12, 13, 14]. In this approach, a graph is constructed for the hypothesized regions of each keyword retrieved in the first pass, in which each node is a hypothesized region, and the edges represent acoustic feature similarity between different regions. Graph re-ranking is based on the concept that a hypothesized region that is strongly connected to other regions with high scores on the graph should have a higher probability of being correct. Thus, in graph-based re-ranking, confidence scores for hypothesized regions propagate over the graph through the edges. In previous research [12], a semi-supervised graph-based re-ranking approach using annotated data has shown good improvements in the IARPA Babel program. However, this approach cannot be applied to keywords that do not occur in the training set. Therefore, for low-resource languages, a fully unsupervised graph-based re-ranking method may be more preferred than the semi-supervised version.

In this paper, we investigate the effectiveness of an unsupervised graph-based re-ranking approach on data from limited language packs of the IARPA Babel program. The experiments on the low-resource languages, Assamese, Bengali and Lao, show that the unsupervised graph-based re-ranking approach improves STD systems using fuzzy matching, and lattices based on different kinds of units including word, subwords, and word/subword hybrids. The rest of this paper is organized as follows. In Section 2, we review the unsupervised graph-based re-ranking approach. In Section 3, we present the details of our STD system used in the IARPA Babel program. We describe our experimental setup and results in Section 4. Finally, we conclude in Section 5.

2. Unsupervised Graph-based Re-ranking using Acoustic Feature Similarity

In this section, we review unsupervised graph-based re-ranking [10, 9]. Given a keyword, a STD system first searches through a spoken archive, returning a set of time spans x that are hypothesized to be the keyword. Each hypothesized region x has a confidence score $C(x)$ that is usually the posterior probability of the keyword, as measured from a lattice [4]. For each keyword, a graph is constructed from the first-pass keyword hit list, as shown in Fig 1, in which each node represents a hypothesized region x of the keyword, and the hypothesized regions that are acoustically similar are connected. The acoustic feature similarity $S(x, x')$ between hypothesized regions x and x' is defined as

$$S(x, x') = 1 - \frac{d(x, x') - d_{min}}{d_{max} - d_{min}}, \quad (1)$$

Supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0013. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government. Hung-yi Lee was supported by the National Science Council of Taiwan under contract numbers NSC 102-2917-I-564-004-A1. We would also like to acknowledge the help of our colleagues at BBN's Speech and Language group for their help with the evaluation.

[1] Here a keyword can refer to a single word or a sequence of words. This definition is used in the Babel program [2].

where $d(x, x')$ is the DTW distance between x and x', and d_{max} and d_{min} are the largest and smallest values of $d(x, x')$ for all pairs of regions in the first-pass hit list. Eq. 1 normalizes the DTW distance $d(x, x')$, and transforms it into the similarity between 0 and 1. The hypothesized regions x and x' are connected if x is among the K-nearest neighbors of x' (based on $S(x, x')$), and if x' is among the K-nearest neighbors of x. If x and x' are connected, $S(x, x')$ would be the weight of the edge between them. After the graph is constructed, a new set of graph-based confidence scores $G(x)$ is obtained, via score propagation on the graph, which can be expressed as

$$G(x) = (1 - \alpha)C(x) + \alpha \sum_{x' \in N(x)} G(x')\hat{S}(x', x), \qquad (2)$$

where $C(x)$ is the original confidence score, α is an interpolation weight between 0 and 1, $N(x)$ is the set of all hypothesized regions connected to x, and x' is a node in $N(x)$. $\hat{S}(x', x)$ is the normalized edge weight $S(x', x)$ over all edges connected to node x' on the graph:

$$\hat{S}(x', x) = \frac{S(x', x)}{\sum_{x'' \in N(x')} S(x', x'')}, \qquad (3)$$

where $N(x')$ is the set of hypothesized regions connected to x'. In Eq. 2, the graph-based score $G(x)$ depends on two factors interpolated by α: the original confidence score $C(x)$ from lattices (the first term on the right hand side of Eq. 2, and score propagation over the graph from all nodes x' connected with x weighted by $\hat{S}(x', x)$ (the second term on the right hand side). In other words, x will have a large graph-based score $G(x)$, or a high confidence based on the graph structure, under the following two conditions. Either x is confident to be a keyword from the ASR system (with high $C(x)$), or x is acoustically similar to other hypothesized regions x' which are confident to be the keyword based on the graph structure (with high $G(x')$). Random walk theory guarantees that a unique set of $G(x)$ satisfying Eq. 2 can be found [15]. Finally, $G(x)$ is combined with $C(x)$ to produce new confidence scores $G'(x)$ for evaluation,

$$G'(x) = C(x)^{1-\delta} G(x)^{\delta}, \qquad (4)$$

where δ is a parameter between 0 and 1.

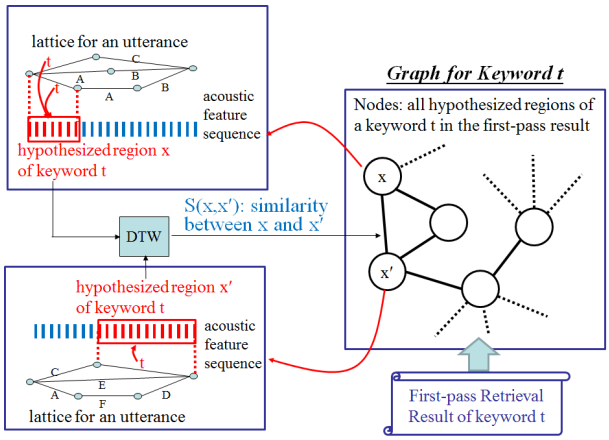

Figure 1: *The graph for each keyword. Each node represents a hypothesized region of the keyword, and the hypothesized regions that are acoustically similar are connected.*

3. Spoken Term Detection (STD) System

Our STD system is described in this section. In Section 3.1, we describe the corpora used, and the ASR system used to generate the lattices. The lattices could consist of words or subword units, as will be described in Section 3.2. In Section 3.3, we describe how the STD system searches through these different lattices, and generates sets of hypothesized regions. The hypothesized regions from different lattices are individually re-ranked by the graph-based approach in Section 2, producing the graph-based confidence scores in Eq. 4. The final results were obtained by merging the overlapped hypothesized regions from different lattices, and summing their weighted graph-based scores to produce the final confidence scores [16].

3.1. Data and Recognition Systems

The audio corpora that we used in our research were from the limited language pack condition of the IARPA Babel program. The recognizers were trained using the Kaldi ASR toolkit [17]. The acoustic features used in speech recognition were tandem features consisting of 13 dimensional speaker-adapted PLP features and stacked Deep Neural Network (DNN) bottleneck features. The stacked DNN bottleneck architecture is a concatenation of two DNNs, where the bottleneck outputs of the first DNN are used as the inputs for the second DNN. Speaker adaptation was also applied to the outputs of the first DNN before feeding it to the second DNN [18]. HMMs were used for acoustic modeling, and discriminative training was done using Minimum Bayes risk (MBR) criterion [19]. The standard Appen lexicons were used, and the language models were created from training data transcripts.

3.2. Generating Subword-based Lattices

In order to address out-of-vocabulary (OOV) terms, word lattices and subword lattices were computed. Since it has been found that integrating the results from lattices based on different kinds of subword units outperforms individual lattices [20, 21, 22], different kinds of subword-based lattices were investigated.

- *Syllable*: Because the phoneme sequence of each syllable, and the mapping between each word and its corresponding syllable sequences were known from the Babel Appen lexicons, a lexicon composed of syllables was directly available. A language model composed of syllable n-grams was trained from transcripts of the training data by transforming the words into syllable sequences. With the syllable-based lexicon and language model, lattices whose hypotheses were syllables could be generated.
- *Word-Syllable Hybrid*: To generate lattices composed of both word and syllable hypotheses, the word lexicon and syllable lexicon were merged to form a word-syllable hybrid lexicon. A hybrid language model was trained by concatenating the conventional word-based transcripts of the training data with a second copy that contained both words and syllable sequences. To ensure that the language model had seen n-grams that include transitions between words and syllables, half of the words in the second copy were randomly selected to be transformed into syllables. The original transcripts and the two copies were used together to train the hybrid language model.
- *Morpheme*: To generate morpheme-based lattices, we employed *Morfessor* [23, 24], a language-agnostic unsupervised system, to do the morphological segmentation. Given the word lexicon as training data, it derived a model which can segment a word into a morpheme sequence. We used

the model to segment all the words in the transcripts of the training data into morphemes. The resulting morphemes were used to create a morpheme lexicon, and the data were used to train a morpheme language model. Pronunciations for morphemes that happened to be a word were extracted directly from the word lexicon. To estimate the pronunciations of the remaining morphemes, we used Sequitur grapheme-to-phoneme (G2P) converter [25] to train a G2P model from the word lexicon, and used the model to estimate the pronunciations of the morphemes which did not happen to be a word. Since G2P is imperfect, we generated M-best pronunciations for each morpheme. Each pronunciation had a weight from the G2P model representing the confidence of correctness, and the weights were normalized such that the summation of the weights of the M pronunciations of a morpheme would be one. All M possible pronunciations of a morpheme and their weights were included in the morpheme lexicon for generating lattices.

- *Word-Morpheme Hybrid*: To generate word-morpheme hybrid lattices, a word-morpheme hybrid lexicon was obtained by merging a morpheme and a word lexicon, and a word-morpheme hybrid language model was trained in the same manner as the word-syllable hybrid language model.
- *Phoneme*: Three sets of phoneme-based lattices were obtained in different ways. One set of phoneme-based lattices was generated directly using a phoneme lexicon and language model composed of phoneme n-grams. The other two sets were transformations of the syllable and morpheme lattices by mapping each syllable or morpheme into its corresponding phoneme sequence.

3.3. Search

In Section 3.3.1, the fuzzy matching [26, 27, 28, 29, 30, 31, 32, 33] used in our system is described. The details of searching different kinds of lattices are described in Section 3.3.2.

3.3.1. Fuzzy Matching

Fuzzy matching is used in spoken term detection to compensate for inevitable ASR errors that will result in missing terms in the word or subword lattices. Requiring an exact match of a multi-word sequence will result in a low recall rate. To address this issue, the time spans of the word sequences which do not match exactly, but are lexically similar to the keyword should also be returned. In the actual implementation, our STD system generates a set of word sequences by substituting and inserting words into the input keyword[2], and uses the set of word sequences to search the word lattices. The confidence scores of the hypotheses found by these word sequences are penalized by multiplying a factor λ smaller than one. For a word sequence obtained by A substitutions, and B insertions from the input keyword, λ would be ${\lambda_s}^A \times {\lambda_i}^B$, where λ_s and λ_i were the respective penalties for substitution and insertion. The search of the word sequence set was efficiently implemented by WFST-based indexing [34]. In our system, there was no limitation on the numbers of substitutions and insertions, but a substitution and an insertion were not allowed to happen consecutively[3], and fuzzy matching was not applied if the original term was a single word.

3.3.2. Searching Subword-based Lattices

When searching subword lattices, the input keyword was transformed into multiple token sequences, where the tokens could be words or subwords. Fuzzy matching was applied based on each token sequence in the manner described in Section 3.3.1, except that the input word sequence in the description of Section 3.3.1 is replaced by a token sequence. The ways of transforming an input keyword into multiple token sequences depended on the particular subword unit, and are described below.

- *Syllable*: To search lattices composed of syllable hypotheses, the entire keyword was first transformed into a phone sequence. G2P would be used to estimate the M most possible phoneme sequences for each OOV word, and if there were n OOV words in a keyword, the keyword would be transformed into M^n phone sequences. For some languages, the syllabic segmentation of a phone sequence is not unique (that is, different syllable sequences can correspond to the same phone sequence). In these cases, all possible syllabic segmentations of the phone sequence were enumerated and used in the fuzzy search[4].
- *Word-Syllable Hybrid*: When searching the word-syllable hybrid lattices, the in-vocabulary (IV) words in the keyword were preserved, but the OOV words were transformed into all possible syllable sequences.
- *Morpheme*: Given a keyword, all possible morphological segmentations were enumerated and used in the fuzzy search.
- *Word-Morpheme Hybrid*: In this case, OOV words in a keyword were transformed into all possible morpheme sequences, while IV words were segmented into morpheme sequences or remained unchanged (both cases were used in search) [5].
- *Phoneme*: To search phoneme-based lattices, a keyword would be transformed into a set of phoneme sequences by the G2P model as mentioned in the *syllable* case.

4. Experiments

The languages considered in this paper were Lao, Assamese, and Bengali from releases IARPA-babel203b-v3.1, IARPA-babel102b-v0.4, and IARPA-babel103b-v0.3, respectively. The number of official evaluation keywords were 3360 for Lao (535 of which were OOV keywords[6]), 3324 for Assamese (739 OOV keywords) and 3352 for Bengali (831 OOV keywords). For better parameter tuning during development, we augmented the official keyword lists with automatically generated keywords [35]. The sizes for the keyword lists used for development were 5132 for Lao (790 OOV keywords), 5053 for Assamese (1298 OOV keywords) and 5053 for Bengali (1488 OOV keywords). For the experiments in each language, we used the standard 10-hour training set from the limited language pack for training acoustic and language models. The size of the dev and eval sets are 10 and

[2]Deletion is not considered in this paper because it was not helpful in our preliminary experiments.

[3]If insertions could happen consecutively, infinitely long word sequences would be generated by inserting an infinite number of words between two words; if substitutions could happen consecutively, by substituting all words in the input keyword, the STD system would lose all information from the input keyword.

[4]Any phone sequence that could not be parsed into a syllabic segmentation was ignored.

[5]A different strategy from the word-syllable hybrid was applied here because preliminary experiments showed that transforming IV words into subword sequences helped the word-morpheme hybrid but hurt the word-syllable hybrid.

[6]Keywords containing one or more OOV words are considered to be OOV keywords.

Table 1: MTWV(%) results on Lao's development set using the augmented keyword list. The columns labelled *First* and *Graph* are respectively the results before and after re-ranking. The rows (A) to (H) are for the results based on different kinds of lattices described in Section 3.2. The results in row (I) are the integration of the results from rows (A) to (E), while the results in row (J) are the integration from rows (A) to (H). The superscripts * indicate that the results after re-ranking are significantly better than the results before re-ranking.

Lao	IV Keywords		OOV Keywords	
	First	*Graph*	*First*	*Graph*
(A): Word	36.87	37.50	3.27	3.65*
(B): Syllable	33.99	35.55*	16.66	17.31
(C): Word+Syllable	36.31	36.41	13.62	15.77*
(D): Morpheme	26.65	28.47*	12.78	13.50
(E): Word+Morpheme	37.26	38.42*	15.11	16.38
(F): Phoneme	25.95	29.51*	13.94	18.53*
(G): Phoneme (from (B))	31.41	33.91*	16.08	19.08*
(H): Phoneme (from (D))	25.87	28.31*	15.70	19.41*
(I):Integrate w/o phoneme	39.34	40.24*	19.34	21.43*
(J):Integrate with phoneme	39.23	40.43*	21.11	24.43*

Table 2: ATWV(%) results on the evaluation sets of Assamese, Bengali and Lao for the official evaluation keyword lists. The rows labelled *First* and *Graph* are respectively the results before and after re-ranking.

	Assamese	Bengali	Lao
First	32.88	26.88	32.67
Graph	34.37	29.30	34.29

75 hours respectively. Tied-state triphone CD-HMMs with 2.5K states, and 18 Gaussian components per state were used for acoustic modeling. A description of the acoustic features used in both speech recognition and re-ranking can be found in [36][7]. BBN's VAD system was used to segment the spoken archive into segments for speech recognition [40]. The WERs on the development set of Lao, Assamese and Bengali were 62.7%, 63.5% and 66.1%, respectively. Before the evaluation, the confidence scores were normalized by Exponential Normalization with Keyword-specific Thresholding [41].

The results on Lao are shown in Table 1. The rows (A) to (H) are for the results based on different kinds of lattices described in Section 3.2. The G2P model generated 10-best pronunciations (or $M = 10$ in Section 3.2 and Section 3.3.2). The results of their integrations are in rows (I) and (J). For integration, the weights of the results from different kinds of lattices were always set to be equal. Fuzzy matching was used, and both λ_s and λ_i in Section 3.3.1 were set to be 0.1. The results of IV and OOV keywords are reported separately in Table 1. The columns labelled *First* and *Graph* are the results before, and after re-ranking, respectively For re-ranking, the parameter K in Section 2 was 10, and α in Eq. 2, and δ in Eq. 4 were both set to be 0.9 to give the graphs larger influence. The superscripts * indicate that the results after re-ranking are significantly better than the results before re-ranking. Significance was performed using a pair-wised t-test with a significance level of 0.05.

We analyze the results in Table 1 by first considering the first-pass results (those shown in columns labeled *First*). Row (A) is for the results of word lattices. Due to fuzzy matching, the MTWV of OOV keywords was not zero, but it was still very poor. Row (B) is for the lattices composed of syllables. Although the performance of syllable was not as good as word-based lattices for IV keywords, syllable-based lattices greatly outperformed word-based lattices for OOV keywords (rows (B) vs. (A)).

The results of the hybrid word/syllable lattices compared with those from syllable-based lattices (rows (C) vs (B)) show that the word/syllable hybrid yielded better performance for IV keywords, but performed worse for OOV keywords. Comparing the results of syllables with morphemes (rows (B) vs (D)) shows that syllable-based lattices outperformed morpheme-based for both IV and OOV keywords. This result indicates that the syllable is very suitable unit for STD on Lao. However, the word-morpheme hybrid performed better than the word-syllable hybrid for both IV and OOV keywords (rows (E) vs (C)). Since a large portion of morphemes were actually words, the number of different n-grams for the word-morpheme hybrid language model was much smaller than the word-syllable hybrid, which makes the language model estimated from the same size of training data more reliable. Therefore, morpheme and word units may be more compatible to each other than syllable.

Row (F) is for phoneme lattices generated directly from a phonemic lexicon and language model, while rows (G) and (H) are phoneme lattices from syllable and morpheme lattices respectively. Among the three sets of phoneme lattices (rows (F), (G) and (H)), the phoneme lattices derived from syllable lattices yielded the best performance on both IV and OOV keywords.

Rows (I) and (J) were the integration without and with phonemes respectively (that is, the results in row (I) are the integration of the results from rows (A) to (E), while the results in row (J) are the integration from rows (A) to (H)). The phoneme lattices improved OOV keywords in integration, but not IV keywords (rows (J) vs (I)).

For the results of re-ranking in Table 1, we found that re-ranking improved the performance no matter what units or types of keywords were used (columns *Graph* vs *First*). Re-ranking was especially helpful for the results of phoneme-based lattices (rows (F) to (H)). The phoneme-based search had larger potential for improvement because the results from phoneme-based lattices were usually noisier due to the lack of lexical constraint. We also observed that, although in the first pass, including the results from phoneme-based lattices for integration was not helpful for IV keywords (rows (J) vs (I) for column *First* of IV keywords), with re-ranking, including phoneme-based lattices benefits the overall system (rows (J) vs (I) for column *Graph* of IV keywords).

Finally, the ATWV(%) results for Assamese, Bengali and Lao on the evaluation sets using the official evaluation keyword lists are shown in Table 2. Due to space limitations, only the ATWVs of the integrated results from all units are reported. We observed that graph-based re-ranking consistently improved the STD performances for the three languages on the evaluation sets.

5. Conclusion

This paper investigated the effectiveness of a graph-based re-ranking approach on three low-resource languages from the limited language packs of the IARPA Babel program. The experiments showed that graph-based re-ranking improved the MTWV and ATWV performance of both IV and OOV keywords on Assamese, Bengali and Lao regardless of units were used for search. In the future, we are planning to explore substitution and insertion penalties that are based on confusion matrices [42], and exploit other information, such as word bursts [43], for re-ranking.

[7]There were some modifications. The filterbank inputs were processed with VTLN warping factors, and Kaldi's pitch extractor [37] and Fundamental Frequency Variation (FFV) features [38] were used instead of Subband Autocorrelation Classification pitch tracker (SAcC) [39].

6. References

[1] *http://www.itl.nist.gov/iad/mig/tests/std/2006/index.html*.

[2] *IARPA broad agency announcement IARPA-BAA-11-02*, 2011.

[3] M. Larson and G. J. F. Jones, "Spoken content retrieval: A survey of techniques and technologies," *Found. Trends Inf. Retr.*, vol. 5, pp. 235–422, 2012.

[4] G. Tur and R. DeMori, *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. John Wiley & Sons Inc, 2011, ch. 15, pp. 417–446.

[5] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, "Automatic speech recognition for under-resourced languages: A survey," *Speech Communication*, vol. 56, pp. 85 – 100, 2014.

[6] C. Parada, A. Sethy, and B. Ramabhadran, "Query-by-example spoken term detection for OOV terms," in *ASRU*, 2009.

[7] C.-P. Chen, H.-Y. Lee, C.-F. Yeh, and L.-S. Lee, "Improved spoken term detection by feature space pseudo-relevance feedback," in *Interspeech*, 2010.

[8] H.-Y. Lee and L.-S. Lee, "Enhanced spoken term detection using support vector machines and weighted pseudo examples," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 6, pp. 1272–1284, 2013.

[9] H.-Y. Lee, P.-W. Chou, and L.-S. Lee, "Improved open-vocabulary spoken content retrieval with word and subword indexing using acoustic feature similarity," in *Special Issue on Information Extraction and Retrieval, Computer Speech and Language*, 2014.

[10] ——, "Open-vocabulary retrieval of spoken content with shorter/longer queries considering word/subword-based acoustic feature similarity," in *Interspeech*, 2012.

[11] Y.-N. Chen, C.-P. Chen, H.-Y. Lee, C.-A. Chan, and L.-S. Lee, "Improved spoken term detection with graph-based re-ranking in feature space," in *ICASSP*, 2011.

[12] K. Audhkhasi, A. Sethy, B. Ramabhadran, and S. S. Narayanan, "Semi-supervised term-weighted value rescoring for keyword search," in *ICASSP*, 2014.

[13] A. Norouzian, R. Rose, S. H. Ghalehjegh, and A. Jansen, "Zero resource graph-based confidence estimation for open vocabulary spoken term detection," in *ICASSP*, 2013.

[14] A. Norouzian, R. Rose, and A. Jansen, "Semi-supervised manifold learning approaches for spoken term verification," in *Interspeech*, 2013.

[15] A. N. Langville and C. D. Meyer, "A survey of eigenvector methods for web information retrieval," *SIAM Rev.*, vol. 47, pp. 135–161, January 2005.

[16] D. Wang, N. Evans, R. Troncy, and S. King, "Handling overlaps in spoken term detection," in *ICASSP*, 2011.

[17] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[18] T. N. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for lvcsr," in *Proc. ICASSP*, 2013.

[19] M. Gibson and T. Hain, "Hypothesis spaces for minimum bayes risk training in large vocabulary speech recognition," in *Proc. InterSpeech*, 2006, pp. 2406–2409.

[20] C.-H. Meng, H.-Y. Lee, and L.-S. Lee, "Improved lattice-based spoken document retrieval by directly learning from the evaluation measures," in *ICASSP*, 2009.

[21] S. wook Lee, K. Tanaka, and Y. Itoh, "Combining multiple subword representations for open-vocabulary spoken document retrieval," in *ICASSP*, 2005.

[22] Y.-C. Pan, H.-L. Chang, , and L.-S. Lee, "Subword-based position specific posterior lattices (S-PSPL) for indexing speech information," in *Interspeech*, 2007.

[23] S. Virpioja, P. Smit, S.-A. Gronroos, and M. Kurimo, "Morfessor 2.0: Python implementation and extensions for morfessor baseline," Aalto University, Tech. Rep., 2013.

[24] M. Creutz and K. Lagus, "Unsupervised discovery of morphemes," in *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning - Volume 6*, 2002.

[25] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, pp. 434 – 451, 2008.

[26] P. Karanasou, L. Burget, D. Vergyri, M. Akbacak, and A. Mandal, "Discriminatively trained phoneme confusion model for keyword spotting," in *Interspeech*, 2012.

[27] R. Wallace, R. Vogt, B. Baker, and S. Sridharan, "Optimising figure of merit for phonetic spoken term detection," in *ICASSP*, 2010.

[28] R. Wallace, B. Baker, R. Vogt, and S. Sridharan, "Discriminative optimization of the figure of merit for phonetic spoken term detection," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, pp. 1677–1687, 2011.

[29] K. Thambiratnam and S. Sridharan, "Dynamic match phone-lattice searches for very fast and accurate unrestricted vocabulary keyword spotting," in *ICASSP*, 2005.

[30] T. Mertens and D. Schneider, "Efficient subword lattice retrieval for german spoken term detection," in *ICASSP*, 2009.

[31] U. V. Chaudhari and M. Picheny, "Improvements in phone based audio search via constrained match with high order confusion estimates," in *ASRU*, 2007.

[32] Y. Y. Tomoyosi Akiba, "Spoken document retrieval by translating recognition candidates into correct transcriptions," in *Interspeech*, 2008.

[33] T. Mertens, D. Schneider, and J. Kohler, "Merging search spaces for subword spoken term detection," in *Interspeech*, 2009.

[34] C. Allauzen, M. Mohri, and M. Saraclar, "General indexation of weighted automata: application to spoken utterance retrieval," in *Proceedings of the Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval at HLT-NAACL 2004*, 2004.

[35] S. Tsakalidis, X. Zhuang, R. Hsiao, S. Wu, P. Natarajan, R. Prasad, and P. Natarajan, "Robust event detection from spoken content in consumer domain videos," in *Interspeech*, 2012.

[36] Y. Zhang, E. Chuangsuwanich, and J. Glass, "Extracting deep neural network bottleneck features using low-rank matrix factorization," in *Proc. ICASSP*, 2014.

[37] P. Ghahremani, B. BabaAli, K. R. D. Povey, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," 2014.

[38] K. Laskowski, M. Heldner, and J. Edlund, "The fundamental frequency variation spectrum," in *Proc. FONETIK*, 2008.

[39] D. Ellis and B. Lee, "Noise robust pitch tracking by subband autocorrelation classification," in *13th Annual Conference of the International Speech Communication Association*, 2012.

[40] T. Ng, B. Zhang, L. Nguyen, S. Matsoukas, X. Zhou, N. Mesgarani, K. Vesely, and P. Matejka, "Developing a speech activity detection system for the DARPA RATS program," in *Interspeech*, 2012.

[41] D. Karakos, R. Schwartz, S. Tsakalidis, L. Zhang, S. Ranjan, T. Ng, R. Hsiao, G. Saikumar, I. Bulyko, L. Nguyen, J. Makhoul, F. Grezl, M. Hannemann, M. Karafiat, I. Szoke, K. Vesely, L. Lamel, and V.-B. Le, "Score normalization and system combination for improved keyword spotting," in *ASRU*, 2013.

[42] G. Chen, O. Yilmaz, J. Trmal, D. Povey, and S. Khudanpur, "Using proxies for OOV keywords in the keyword search task," in *ASRU*, 2013.

[43] J. Chiu and A. Rudnicky, "Using conversational word bursts in spoken term detection," in *Interspeech*, 2013.