A Keyword-Aware Language Modeling Approach to Spoken Keyword Search

I-Fan Chen[1] · Chongjia Ni[2] · Boon Pang Lim[2] · Nancy F. Chen[2] · Chin-Hui Lee[1]

Received: 13 November 2014 / Revised: 16 February 2015 / Accepted: 23 March 2015 / Published online: 21 April 2015

Abstract A keyword-sensitive language modeling framework for spoken keyword search (KWS) is proposed to combine the advantages of conventional keyword-filler based and large vocabulary continuous speech recognition (LVCSR) based KWS systems. The proposed framework allows keyword search systems to be flexible on keyword target settings as in the LVCSR-based keyword search. In low-resource scenarios it facilitates KWS with an ability to achieve high keyword detection accuracy as in the keyword-filler based systems and to attain a low false alarm rate inherent in the LVCSR-based systems. The proposed keyword-aware grammar is realized by incorporating keyword information to re-train and modify the language models used in LVCSR-based KWS. Experimental results, on the *evalpart1* data of the IARPA Babel OpenKWS13 Vietnamese tasks, indicate that the proposed approach achieves a relative improvement, over the conventional LVCSR-based KWS systems, of the actual term weighted value for about 57 % (from 0.2093 to 0.3287) and 20 % (from 0.4578 to 0.5486) on the limited-language-pack and full-language-pack tasks, respectively.

✉ I-Fan Chen
ifanchen@gmail.com

Chongjia Ni
nicj@i2r.a-star.edu.sg

Boon Pang Lim
bplim@i2r.a-star.edu.sg

Nancy F. Chen
nfychen@i2r.a-star.edu.sg

Chin-Hui Lee
chl@ece.gatech.edu

[1] School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA

[2] Institute for Infocomm Research, Singapore, Singapore

Keywords Keyword spotting · Keyword search · Filler · Spoken term detection · Grammar network · LVCSR

1 Introduction

Spoken keyword search (KWS) [1, 2] is a task of detecting a set of preselected keywords in continuous speech. The technology has been used in various applications, such as spoken term detection [3–6], spoken document indexing and retrieval [7], speech surveillance [8], spoken message understanding [9, 10], etc. In general, KWS systems can be categorized into two groups: classic keyword-filler based [1, 2] and large vocabulary continuous speech recognition (LVCSR) based KWS [3–6].

In classic keyword-filler based KWS, speech inputs are treated as sequences of keywords and non-keywords (often referred to as fillers) [1, 2]. It performs keyword search by decoding input speech into keywords and fillers with time boundary information. To do so, for each keyword in the system a corresponding keyword model is established for modeling its acoustic properties, while all non-keywords share a filler acoustic model. The decoding grammar[1] is a simple keyword-filler loop grammar (as shown in Fig. 1a). Because of its simplicity, a keyword-filler based system requires only a small amount of training data to obtain a reasonable performance. But the system can only be used for detecting a small set of predefined keywords.

In the 90s, with the rapid increase in computing power and data resources [11, 12], implementing an LVCSR system with

[1] In this study, a grammar is defined as a search graph or network whose paths from the initial to final nodes represent valid word sequences in a system with corresponding scores, and the graph/network is easily realized by weighted finite-state automata (WFSA) [40, 41].

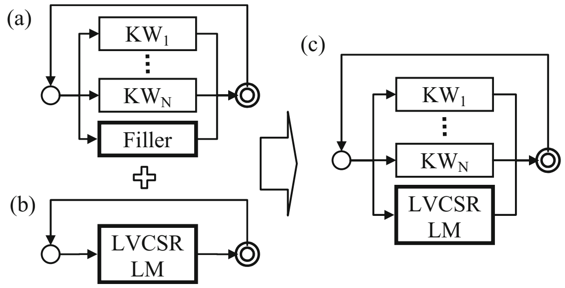


Figure 1 **a** Grammar of classic keyword-filler based KWS, **b** LM-based grammar used by LVCSR-based KWS, and **c** the proposed keyword-aware grammar, which combines the grammars used in the two KWS frameworks.

a good performance was no longer impractical. LVCSR systems became mainstream in KWS research [4, 5, 13, 14] on languages with rich training resources, e.g., English, Arabic, and Mandarin Chinese. LVCSR-based systems solve the keyword search problem from another aspect. Instead of decoding the input speech into a sequence of keywords and fillers, they convert input speech into general text documents using speech-to-text (STT) techniques with language model (LM) [15] based grammar [3–5]. These text documents can be in different formats, such as *N*-best sentences or lattices generated by the LVCSR systems at word [3–5] or sub-word (e.g., syllable [16] or phone [3, 6]) levels. Since they can be used for searching any keyword, LVCSR-based KWS is more flexible than conventional keyword-filler based KWS on keyword targets because the relationship between keywords and non-keywords are better characterized in an *n*-gram based LM, which play a key role in determine the system performance. However, a high-performance LM typically requires a significant amount of text training data [11, 17] which makes it a major performance bottleneck for LVCSR-based KWS in resource-limited applications. This is especially an issue when LMs can only be built using transcribed speech data as in the recent Babel program [18] sponsored by IARPA (Intelligence Advanced Research Projects Activity) of the United States.

Recently, KWS under limited-resource conditions [19–23] has become a research focus because training data collection is often one of the most time-consuming and expensive efforts in the overall system building process. While there are more than thousands of languages in the world [24] recorded in many different conditions, it is usually not practical for KWS system designers to collect and transcribe a great amount of training speech data for every language of interest in a particular environment. In most cases, for a new language in a specified acoustic condition, there would only be a very limited amount of training data available for system training.

Various techniques were therefore brought out to enhance the KWS performance under limited-resource conditions. Indirect approaches, such as using more robust or informative acoustic features (e.g., bottleneck features [25], tonal features [26]), keyword verification [27, 28], and system combination [29], which enhance the KWS performance without tackling the limited-resource modeling problems have been shown to achieve reasonable performance improvement. Techniques directly addressing the modeling problems under resource-limited conditions are also proposed by many research groups. For example, data-augmentation methods such as semi-supervised training [30], acoustic data-perturbation [31], cross-lingual transfer learning [32] are shown effective to improve acoustic models for limited-resource languages. However, despite the great amount of works for the enhancement of limited-resource KWS, there are relatively few researches focusing on language modeling in this newly emerged research field. To complete this missing piece, in this study, the language modeling problem for resource-limited KWS is specifically analyzed and studied.

The research paradigm shift toward the limited-resource conditions inspires us to revisit classic keyword-filler based KWS since such systems often perform well in low-resource conditions. If the keyword-filler grammar and *n*-grams in LVCSR can be unified, the integrated system is expected to achieve higher performance than either type of the two KWS systems. In this study, we propose a keyword-aware grammar [33, 34] to combine these two frameworks. Experimental results indicate that the proposed grammar is flexible as in LVCSR-based KWS and able to achieve a significant improvement over the two conventional systems on the KWS system performance regardless of the amount of system training resources.

The rest of the paper is organized as follows. In Section 2, conventional keyword-filler and LVCSR-based KWS are presented. A potential keyword prior underestimation issue caused by limited LM training data for LVCSR-based KWS is also highlighted. The keyword-aware language modeling approach, alleviating the prior underestimation problem, is then proposed in Section 3. Three realizations for the proposed grammar are then presented in Section 4. Next the experimental setup is detailed in Section 5, and the experimental results are analyzed in Section 6. Finally we conclude our findings with future work in Section 7.

2 Spoken Keyword Search Problem

Spoken keyword search is an application of the automatic speech recognition (ASR) technology that focuses on the recognition of keywords. Given a speech utterance *O* and a text-based query *q*, a KWS system detects the query *q* in the utterance by finding the best *term* sequence, W^*, corresponding to the utterance *O* as follows:

$$W^* = \underset{W}{\operatorname{argmax}} P(W|O). \qquad (1)$$

If the query, q, does exist in the utterance, then we expect $W^*=h\cdot q\cdot f$, where h and f are *term* sequences (which we do not really care) preceding and following the query in the utterance, and "·" is a concatenation operator. Otherwise a detection miss error occurs. Note that usually miss errors are considered more serious than false alarms in KWS since the later can still be removed with a further utterance-verification stage [35–38].

With Bayes' rule, Eq. (1) can be rewritten as

$$W^* = \underset{W}{\operatorname{argmax}}\, P(O|W)P(W), \tag{2}$$

where P($O|W$) is the likelihood of the utterance, O, given the hypothesized *term* sequence, W; P(W) is the prior probability for the hypothesized *term* sequence. In general, the likelihood, P($O|W$), can be computed with acoustic models, and P(W) is modeled by system language models. Equation (2) can then be solved by Viterbi beam search to alleviate the computational burden caused by the large search space. Note in many applications, instead of using only the 1-best result, W^*, lattices or N-best sentences with confidence scores can also be generated for keyword detection [3–5, 39]. Thus for an utterance containing a query, q, it is a key to make sure the hypothesized *term* sequences, $W=h\cdot q\cdot f$, containing the query have probabilities high enough to stay in the search beam and be preserved in the final lattices or N-best sentences. More precisely, the probabilities of P($q|h$) estimated by language models should be sufficiently high to linguistically allow the query-containing search path to be retained in the beam width when processing the speech segment of the query in the utterance. Otherwise, the query would be missed.

The two conventional KWS groups utilize a similar acoustic modeling approach, but they are very different in the definition of *terms* and the estimation of the prior probability, P(W). The differences in their language modeling approaches lead to their contrastive performance characteristics as explained in the following sections.

2.1 Keyword-Filler Based KWS

In a standard keyword-filler based KWS system, the *terms* are defined as a set of keywords and filler (representing all non-keywords). The probability of each *term* in the utterance is usually assumed to be context independent in the standard keyword-filler loop grammar (shown in Fig. 1a), namely P($q|h$)=P(q). And it is often assumed that P(q) is a uniform distribution over all *terms* and thus equal to 1/N, where N is the number of *terms* in the system. For most keyword-filler based KWS systems, N is a number smaller than 100 [1, 37–39]. Since the prior probabilities for most keywords are less than 10^{-4} in practical settings,[2] by assuming P(q)=1/N≥1/100≫ 10^{-4}, the estimation of P($q|h$)=P(q) in standard keyword-filler based KWS is linguistically sufficient to preserve the keyword in the search path in most cases. As a result the systems usually achieve a high detection rate despite the over-estimated priors sometimes create a great amount of false alarms as well.

2.2 LVCSR-Based KWS

In LVCSR-based KWS, n-gram is used for evaluating P($q|h$). Given an L-word query, $q=(w_1,w_2, \ldots,w_L)$, the conditional probability of q given h is evaluated as

$$P\left(q\middle|h\right) = P\left(w_1 w_2 \cdots w_L\middle|h\right) \cong \prod_{i=1}^{L} P_{n\text{-}gram}\left(w_i\middle|h_i\right), \tag{3}$$

where $P_{n\text{-}gram}(.)$ is the probability estimated by the system n-gram LM, and h_i is the history of w_i in the query q dictated by the order of the n-gram LM. This prior estimation helps LVCSR-based KWS achieve better detection accuracy than keyword-filler based KWS when sufficient LM training data is available [39].

2.3 Prior Underestimation in LVCSR-Based KWS

Equation (3) shows how the conditional keyword priors, P($q|h$), are evaluated in the LVCSR-based KWS framework using n-gram LMs. However, in resource-limited tasks the amount of LM training text is often insufficient to cover keyword-related domains and causes extremely low estimates for the n-gram probabilities of the keywords. In other words, a potential problem for LVCSR-based KWS is that the keyword prior probabilities, P($q|h$), might be underestimated by Eq. (3) due to domain mismatch resulting in a high miss rate for the keywords. The problem is more pronounced for multi-word keywords with a large L because of the compound probability multiplications.

3 Keyword-Aware Language Modeling

When the system n-gram LMs are trained with limited or topic-mismatched data, LVCSR-based KWS suffers from the abovementioned prior underestimation problem leading to a high miss rate in KWS. To alleviate the situation, we propose a keyword-aware language modeling approach which integrates the prior estimation in keyword-filler based KWS into the LVCSR-based KWS framework for an accurate evaluation of the keyword priors.

[2] For example, Fig. 4 shows the averaged keyword prior probabilities in the IARPA Babel Vietnamese data [32] are in the range of 5×10^{-5} to 5×10^{-6}.

As in LVCSR-based KWS, the proposed keyword-aware KWS framework also utilizes an underlying LVCSR system but with keyword priors computed by:

$$P_{KW-aware}\left(q\middle|h\right) = \max\left\{P_{n-gram}\left(q\middle|h\right)\ ,\quad \kappa\right\}, \tag{4}$$

where κ is a parameter for query q to control the minimum keyword prior value allowed in the system. Note that if we set κ to 0, Eq. (4) would become Eq. (3), which is LVCSR-based KWS. When setting κ to $1/N$ for an *N-keyword* task, Eq. (4) becomes the prior used in the keyword-filler based KWS since in most cases $1/N$ is larger than $P_{n\text{-gram}}(q)$. The two conventional KWS frameworks therefore can be seen as special cases of the proposed framework. By tuning the parameter κ for each query in the system, we are able to adjust the sensitivity of a system to the keywords of interest even when the *n*-gram LM of the system is not well trained.

The proposed keyword-aware framework also preserves the keyword flexibility because of the underlying LVCSR system. New keywords can be searched in the transcribed documents of the proposed system without reprocessing the speech signal. Note that in the keyword-aware LM only the prior probabilities of the preselected keywords are modified, while the rest of the *n*-gram probabilities in the original LM remain the same. The transcribed document of the proposed system is therefore exactly the same as the original LVCSR-based KWS system for regular *terms* in the system vocabulary. As a result, performances of the new keywords, whose prior probabilities are not modified, would be similar to the original LVCSR-based KWS.

The proposed LM grammar can be realized in a weighted finite-state transducer (WFST) based LVCSR system [40] by directly inserting additional keyword paths to the *n*-gram based grammar WFSA [41] of the system to form a *keyword-aware (KW-aware) grammar* WFSA as illustrated in Fig. 1c. However since the word sequence of a keyword can be present in both paths for the language model and keywords, extra caution is required to ensure the WFSA is deterministic and can be minimized. For rapid-prototyping, instead of performing KWS with complex grammar-level WFSAs, in the next section we propose three methods that approximate the effect of the proposed LM approach by adjusting the probabilities of keywords in the *n*-gram language models used by the LVCSR-based KWS systems. The proposed LMs can be easily implemented in any state-of-the-art LVCSR-based KWS systems.

4 Realization of the KW-Aware Grammar

4.1 Keyword-Boosted Language Model

The most straight-forward way to boost the probability of the word sequences of keywords in a language model is adding the keywords to the training text of the language model. Given the training data for the language model and a list with *N* target keywords, we append each keyword to the training text *k* times. The resulting training text for the language model will be the original training transcriptions with additional *N*·*k* lines of keywords. The parameter *k* which indicates the number of times a keyword repeat in the training text is a parameter to be tuned. We call this a keyword-boosted LM (KW-boosted LM); [19] has explored similar methods and showed it help improve system performance on Cantonese KWS tasks.

Language models trained by this keyword-appended text will have a higher probability for the word sequences of keywords and thus are more sensitive to the predefined keywords even when the original training text contains very little information about them.

4.2 Keyword Language Model Interpolation

The KW-boosted LM approach adjusts the probabilities of keyword paths to the other paths in the original language model by setting the repetition number *k* of the keywords in the training text. However, since *k* can be any positive integer, such an infinite range of possibilities makes it difficult to optimize system performance. To alleviate this problem, instead of appending keywords to the original LM training text, we train a keyword language model using keyword text alone and then perform a linear interpolation with the original language model using Eq. (5). We call this keyword language model (KWLM) interpolation.

$$P_{INT\_LM}\left(w\middle|h\right) = \alpha\cdot P_{KWLM}\left(w\middle|h\right) + (1-\alpha)P_{LM}\left(w\middle|h\right) \tag{5}$$

In Eq. (5), the $P_{\mathrm{INT\_LM}}(w|h)$ is the interpolated probability between the keyword LM and the original LM for the *n*-gram (h, w), where h is the history and w is the current word. Note that in the proposed KWLM interpolation, the parameter α, which tunes the weight of keyword LM to the original LM in the final LM, is in a manageable range of [0,1] instead of the open range [0,∞). In addition, it makes linguistic sense to keep the two text lists separate as they are from intrinsically different sources. Integrating the two text lists via an interpolation weight makes the solution more elegant than the previous approach.

4.3 Context-Simulated Keyword Language Model (CS-KWLM) Interpolation

In the keyword language model training text, each keyword is treated as an individual sentence as shown in Fig. 2a. This makes the keyword language model overemphasize the probability of the keyword appearing at the beginning and the end of a sentence. To remove this bias, in the context-simulated keyword language model training text we put context terms

(a)		(b)			
	keyword_1		ctx-terms	keyword_1	ctx-terms
	keyword_2			…	
	keyword_3		ctx-terms	keyword_2	ctx-terms
	…			…	
	keyword_N		ctx-terms	keyword_N	ctx-terms

Figure 2 Illustration of the training text for (**a**) KWLM, and (**b**) context-simulated keyword language model (CS-KWLM).

before and after each keyword to simulate the situation where keywords are embedded in real sentences. Figure 2b illustrates the training text for CS-KWLM. The context terms can be selected as bigrams or trigrams with high probabilities[3] in the original language model. Once the context-simulated keyword language model is trained, we can use Eq. (5) to obtain another interpolated language model which approximates the proposed keyword-aware grammar for KWS.

5 Experimental Setup

Experiments were conducted on the IARPA Babel OpenKWS13 Vietnamese limited language pack (LLP) and full language pack (FLP) tracks [42], while we put more emphasis on the more-challenging LLP task in this paper. The training set of the FLP task consists of 80 h of transcribed audio; the LLP task shares the same audio training data but only a 10-h transcription subset are allowed to be used. The audio data is conversational speech between two parties over a telephone channel, which can be landlines, cellphones, or phones embedded in vehicles, with the sampling rate set at 8000 Hz. For system tuning, a 2-h subset of the IARPA development set (denoted as dev2h in this paper) was used to speed up the tuning process.

The 15-h evaluation part 1 data (released as *evalpart1* by NIST) was used for testing. The keyword list contains 4065 phrases including out-of-vocabulary words not appearing in the training set. The performance of keyword search was measured by the Actual Term Weighted Value (ATWV) [13]:

$$ATWV = 1-\frac{1}{K}\sum_{kw=1}^{K}\left(\frac{N_{Miss}(kw)}{N_{True}(kw)}+\beta\frac{N_{FA}(kw)}{T-N_{True}(kw)}\right), \qquad (6)$$

where K is the number of keywords, $N_{Miss}(kw)$ is the number of true keyword tokens that are not detected, $N_{FA}(kw)$ is the number of false alarms, $N_{True}(kw)$ is the number of keywords in reference, T is the number of seconds of the evaluation audio, and β is a constant set as 999.9. Note that the IARPA Babel program set ATWV=0.3 as the benchmark for the Vietnamese KWS task.

All keyword search systems were LVCSR-based[4] with hybrid DNN-HMM acoustic models built with the Kaldi toolkit [43]. In fact, readers can easily reproduce all baseline results presented in this paper by running the Babel recipe provided in the Kaldi toolkit. The DNNs were trained with sMBR sequential training [44]. The acoustic features were bottleneck features appended with fMLLR features, while the bottleneck features were built on top of a concatenation of PLP, fundamental frequency (F0), and fundamental frequency variation (FFV) features. For the LLP task, since some items on the keyword list were out-of-vocabulary (OOV) words, we used a grapheme-to-phoneme (G2P) approach [45] to estimate the pronunciation for those OOV words. They were then merged into the original LLP lexicon provided by IARPA to form the system lexicon.

The LLP baseline language model is a trigram LM trained with the transcriptions of the 10-h training text. Since the amount of the training data was very limited, lots of keywords and key phrases were unseen to the language model and therefore they resulted in very low estimated probabilities in the decoding phase. Table 1 shows how serious the problem is. In the first row of Table 1, there were 3275 out of the 4065 keywords unseen in the training text, namely *n*-grams used by these terms ended up with low probabilities in the baseline language model. Moreover, there were 619 keywords consisting of out-of-vocabulary words, which means that the baseline language model will give these terms nearly zero in backing-off probability and make them easily pruned away during decoding. Therefore, it is not surprising that a substantial amount of keywords will be missed if the baseline language model was used for decoding. This is why we need the keyword-aware language models to alleviate the problem.

6 Experimental Results and Discussion

6.1 OpenKWS13 Limited Language Pack Task

We first tuned parameters of the three keyword-aware (KW-aware) systems on the dev2h subset. The parameter k of the KW-boosted LM method was empirically[5] set to 5 without fine-tuning to save development time since the range for the selection is quite wide. Table 2 compares performance of different systems on the dev2h data. Note that the Babel OpenKWS13 Vietnamese data is relatively difficult when

[3] In this research, we used all the bigrams in the original Kneser-Ney smoothed LM as context terms.

[4] We have obtained very poor performances (negative ATWVs) for keyword-filler based KWS systems due to an extremely large amount of false alarms caused by the noises in the test data. Therefore keyword-filler based KWS systems were not considered here.

[5] We observed that by adjusting k=5 the performance is significantly better than setting k=1 as in [19]. However, the differences became trivial when k is larger than 5.

Table 1 Numbers of terms unseen in the training data and terms containing OOV words among the given list of 4065 keywords and key phrases in the LLP task.

	#Keywords	Percentage in the keyword list
Terms unseen in training data	3275	80.6 %
Terms containing OOV words	619	15.2 %

Table 3 WER (in %) and ATWV performance of LLP systems with different language models on the *evalpart1* data.

LLP Systems (evaluated on *evalpart1*)		WER	ATWV
Baseline LM		65.0	0.2093
KW-aware LM	KW-boosted LM (k=5)	65.1	0.2715
	KWLM Interpolation (α=0.6)	66.7	0.3186
	CS-KWLM Interpolation (α=0.6)	66.0	**0.3287**

compared to most of the commonly used datasets. Despite using the state-of-the-art LVCSR techniques, the Kaldi baseline system still had a very high word error rate (WER) and could only achieve 0.2265 of ATWV (first row in Table 2). For the KW-boosted LM system, even without fine-tuning, the method brought a 26 % relative gain on the ATWV already. The slight WER improvement over the baseline system is due to the additional n-gram information provided by the extra appended keyword text in the LM training data. For KWLM and CS-KWLM systems, after tuning the best α in Eq. (5) to be 0.6 for both systems the ATWVs improved to 0.3431 and 0.3546, respectively.

Table 3 shows the experiment results on the *evalpart1* data. A very similar trend of system performance on the dev2h data is observed. The ATWV of the Kaldi baseline was only 0.2093, which is still far below the IARPA Babel program's minimal requirement. The KW-boosted LM significantly reduced this performance gap and reached the ATWV of 0.2715. By adopting KWLM and CS-KWLM interpolation methods, our systems successfully achieved the goal of the program. For the CS-KWLM system, which had the best ATWV performance, the overall ATWV improvement over the baseline system is 0.1194 absolute and more than 50 % relative. Note that optimizing system ATWV over the evaluation keywords using the proposed methods does not hurt WER performance of the underlying LVCSR systems significantly. In other words, the lattices generated by the proposed systems still have similarities for non-keyword terms to the lattices generated by the baseline system. Therefore, even when adding new keywords which are not in the current list for evaluation, in the worst case, the proposed system would have a similar performance to the baseline system for those new keywords.

Table 2 WER (in %) and ATWV comparison of LLP systems with different language models on the dev2h data.

LLP Systems (evaluated on dev2h)		WER	ATWV
Baseline LM		62.5	0.2265
KW-aware LM	KW-boosted LM (k=5)	62.3	0.2853
	KWLM Interpolation (α=0.6)	64.2	0.3431
	CS-KWLM Interpolation (α=0.6)	63.5	**0.3546**

6.1.1 Comparison of KWLM and CS-KWLM Interpolation

The major difference between KWLM and CS-KWLM is the introduction of the context information derived from the original LM. In Fig. 3, the ATWVs of the two systems with different α on the dev2h data were compared. For α smaller than 0.6, the CS-KWLM system outperformed the KWLM system by more than 0.02 ATWV consistently. This demonstrated that the context information provides the CS-KWLM interpolated LM a better connectivity between the keyword LM and the original LM. In other words, it makes the CS-KWLM approach better represents the keyword-aware grammars.

Both systems reach the highest ATWV value when α=0.6. The ATWV of the CS-KWLM system starts dropping fast when α gets larger than 0.6 because of the increased false alarms. However, as long as α is tuned with a representative development data, the risk of such increase in false alarms is small since the optimal α is quite consistent as observed in Tables 2 and 3.

6.1.2 ATWV Analysis for IV and OOV Keywords

In Table 4 we compared ATWV of in-vocabulary (IV) and out-of-vocabulary (OOV) keywords for the baseline and the CS-KWLM systems. Note that for the OOV queries, the

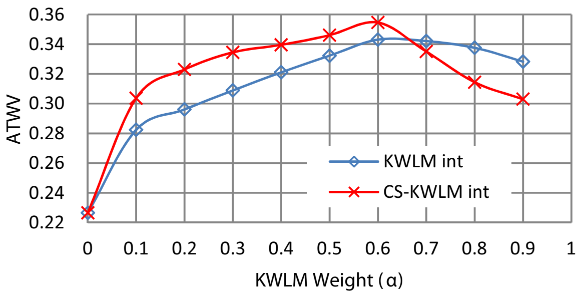


Figure 3 ATWV on dev2h with different keyword LM weights α for both KWLM and CS-KWLM interpolation methods.

Table 4 ATWV performance of all, in-vocabulary (IV), and out-of-vocabulary (OOV) queries for the baseline LM and CS-KWLM Interpolation systems on the *evalpart1* data.

LLP Systems (*evalpart1*)	all	IV	OOV
Baseline LM	0.2093	0.2338	0.0924[a]
CS-KWLM Interpolation (α=0.6)	0.3287	0.3485	0.2343

[a] It was not zero because the baseline also used the G2P lexicon in Section 5 for a fair comparison with the KW-aware systems

baseline had a very low ATWV because those queries are represented with nearly zero probabilities in the language model, causing a high miss error rate. By using the CS-KWLM method to alleviate this problem, ATWV for the OOV queries achieved 0.2343, which is a 154 % relative improvement. For the IV queries, the CS-KWLM method also brought a relative ATWV improvement of 49 %. Therefore, the proposed approach is effective for keywords in both categories, especially for OOV keywords.

6.1.3 ATWV for Seen and Unseen Keywords

When dealing with topics not well-observed, data mismatch is assumed to be a major cause of prior probability underestimation in *n*-gram training. We next compared performances of seen and unseen keywords in the LM training set in the LLP task. The unseen keywords can be viewed as keywords whose topics were not covered by the training data. In other words, even IV keywords might still be unseen to the system LM. Because there were only 10-h transcriptions available for LM training in the LLP task, 3275 out of the 4065 keywords (see Table 1) were unseen to the baseline *n*-gram LM. In other words, more than three quarters of the evaluation keywords suffered the mismatch issue in the *n*-gram LM.

In Table 5, for both keyword groups the proposed KW-aware system showed increased ATWVs in both cases. The improvement is especially significant for unseen keywords – about a 0.15 absolute (from 0.2 to 0.35, 75 % relative) ATWV increase over the baseline. Furthermore, the small improvement for the seen keywords showed that their priors might still be underestimated even for keywords already appearing in the LM training set and needed to be adjusted with the proposed method.

Table 5 ATWV for seen and unseen keywords in the LLP task.

LLP Systems (*evalpart1*)	all	seen	unseen
Baseline LM	0.2093	0.2350	0.1985
CS-KWLM Interpolation (α=0.6)	0.3287	0.2648	0.3574

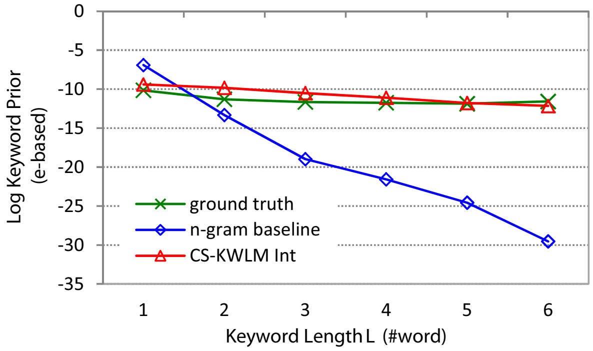


Figure 4 The ground-truth of log keyword priors and log keyword priors estimated by *n*-gram baseline and KW-aware LM on *evalpart1* data for keywords with different lengths, *L*.

6.1.4 Priors Estimation for Keywords in Different Lengths

In Section 2.3, it is claimed that the underestimation problem is more pronounced for multi-word keywords with large *L* because of the compound probability multiplication in *n*-gram LMs. We show some evidence for the statement here. Figure 4 displays the average log priors for keywords with different lengths, *L*, and compares the priors estimated by the two systems on the *evalpart1* data. For each keyword appearing in the *evalpart1* data, its ground-truth prior was estimated by dividing the keyword occurrence count with the total word count in the dataset. In Fig. 4, the “ground-truth” log keyword priors in the *evalpart1* data remained in the range of −10 to −12 for all the keyword lengths evaluated.

The estimated keyword priors for the two systems were evaluated by searching the best keyword path in the system decoding grammar WFSA for each keyword. The weight of the best path was used as the estimated prior for the keyword in the systems. In Fig. 4, for the *n*-gram baseline system, though the estimated keyword priors was quite close to the real values for single-word keywords, the priors were seriously underestimated for longer keywords. The curve of the *n*-gram system monotonically decreased as the keyword length increased. It is clear that the *n*-gram baseline estimated priors are seriously underestimated for keyword with length $L \geq 3$. For

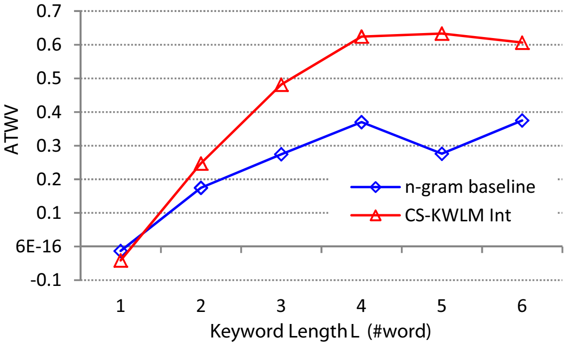


Figure 5 ATWV for keywords with different length, *L*, of the baseline and KW-aware systems on the *evalpart1* data.

Table 6 ATWV of the FLP systems with original *n*-gram and KW-aware LMs on *evalpart1* data.

FLP Systems (*evalpart1*)	ATWV
Baseline LM	0.4578
CS-KWLM Interpolation (α=0.7)	**0.5486**

example, the system underestimated the prior probabilities at the scale of 5×10^8 for the 6-word keywords. The underestimation problem was alleviated by the proposed methods. By boosting prior probabilities for each keyword with CS-KWLM, the prior estimation of the KW-aware system is very close to the real priors regardless of the number of words in a key phrase.

6.1.5 ATWV for Keywords with Different Lengths

To verify if the underestimation problem of the *n*-gram LM for keyword prior estimation is the major factor affecting KWS performances, we further compared the two systems on keywords with different lengths. Figure 5 displays the ATWV curves for the *n*-gram baseline and the KW-aware systems in the LLP task. In general, a KWS system has better detection performance for longer keywords because more acoustic context information is available for the system to make correct decisions. However, because of the misses caused by the underestimated keyword priors, the ATWVs of the *n*-gram baseline system in Fig. 5 only increased slowly with the keyword lengths. On the other hand, the ATWV curve for the KW-aware system has a clear improvement over the baseline system, and the improvement is especially larger for longer keywords. For example, the KW-aware system successfully detected two out of the three five-word keyword, "đăng ký mùa hè xanh", in the evaluation data without any false alarm, while the *n*-gram baseline system missed all of them. The KW-aware system showed a similar ATWV to the *n*-gram baseline on single-word keywords because priors of them were not as seriously underestimated due to LM smoothing [46].

6.2 OpenKWS13 Full Language Pack Task

Our last experiment verifies if the proposed language modeling approach works even when more system training data are available. Table 6 shows the performance of FLP systems on the *evalpart1* data. With more training data, the baseline system achieved the program goal with an ATWV of 0.4578. However, the performance could be further improved substantially (20 % relative) by adopting the CS-KWLM interpolation method. This result shows that the underestimation problem does not go away by simply increasing the amount LM training data, and the proposed keyword-aware language modeling is an effective solution providing significant performance enhancement irrespective to the amount of system training resources.

7 Conclusion

In this paper, we propose a keyword-aware language modeling approach to combine the advantages of the conventional keyword-filler based KWS and the LVCSR-based KWS systems. For rapid-prototyping, three methods that approximate the effect of the keyword-aware grammar are investigated. Results on the IARPA Babel OpenKWS13 Vietnamese LLP and FLP tasks showed that the proposed keyword-aware method is effective in alleviating the prior underestimation problem of LVCSR-based KWS, especially for long and unseen keywords. It also significantly improved the ATWV performance regardless of the amount of system training resources. We are now working on discriminative criteria for the proposed keyword-aware grammar by only boosting keyword priors when needed and suppressing the overestimated priors in the original LM to reduce unwanted false alarms.

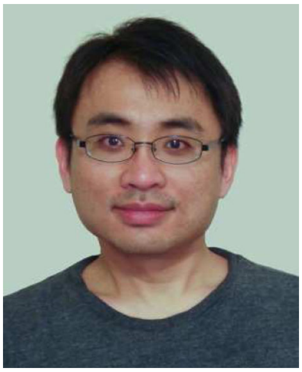
Acknowledgments This study uses the IARPA Babel Program Vietnamese language collection release babel107b-v0.7 with the LimitedLP and FullLP training sets.

References

1. Wilpon, J. G., Rabiner, L. R., Lee, C.-H., & Goldman, E. (1990). Automatic recognition of keywords in unconstrained speech using hidden Markov models. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 38*(11), 1870–1878.
2. Rose, R. C., & Paul, D. B. (1990). A hidden Markov model based keyword recognition system. In *Proceedings of ICASSP*, Albuquerque, NM (vol. 1, pp. 129–132): IEEE. doi:10.1109/ICASSP.1990.115555
3. Vergyri, D., Shafran, I., Stolcke, A., Gadde, R. R., Akbacak, M., Roark, B., et al. (2007). The SRI/OGI 2006 spoken term detection system. In *Proceedings of Interspeech*, (pp. 2393–2396): ISCA. http://www.isca-speech.org/archive/interspeech_2007/i07_2393.html
4. Mamou, J., Ramabhadran, B., & Siohan, O. (2007). Vocabulary Independent spoken term detection. In *Proceedings of SIGIR* (pp. 615–622): ACM. doi:10.1145/1277741.1277847
5. Miller, D. R., Kleber, M., Kao, C.-l., Kimball, O., Colthurst, T., Lowe, S. A., et al. (2007). Rapid and accurate spoken term detection. In *Proceedings of Interspeech*: ISCA. http://www.isca-speech.org/archive/interspeech_2007/i07_0314.html
6. Wallace, R., Vogt, R., & Sridharan, S. (2007). A Phonetic Search Approach to the 2006 NIST Spoken Term Detection Evaluation. In *Proceedings of Interspeech*: ISCA. http://www.isca-speech.org/archive/interspeech_2007/i07_2385.html
7. Makhoul, J., Kubala, F., Leek, T., Liu, D., Nguyen, L., Schwartz, R., et al. (2000). Speech and langauge technologies for audio indexing and retrieval. *Proceedings of the IEEE, 88*(8), 1338–1353.
8. Warren., R. L. (2001). Broadcast speech recognition system for keyword monitoring. U.S. Patent 6332120 B1. http://www.google.tl/patents/US6332120
9. Kawahara, T., Lee, C.-H., & Juang, B.-H. (1998). Key-phrase detection and verification for flexible speech understanding. *IEEE Transactions on Speech and Audio Processing, 6*(6), 558–568.
10. Juang, B.-H., & Furui, S. (2000). Automatic recognition and understanding of spoken language – a first step toward natural human-machine communication. *Proceedings of the IEEE, 88*(8), 1142–1165.

11. Rosenfeld, R. (2000). Two decades of statistical language modeling: where do we go from here? *Proceedings of the IEEE, 88*(8), 1270–1278.
12. Pallett, D. S. (2003). A look at NIST's benchmark ASR tests: past, present, and future. In *Proceedings of ASRU*, (pp. 483–488): IEEE. doi:10.1109/ASRU.2003.1318488
13. Fiscus, J. G., Ajot, J., Garofolo, J. S., & Doddintion, G. (2007). Results of the 2006 spoken term detection evaluation. In *Proceedings of SIGIR*: ACM. http://www.itl.nist.gov/iad/mig/publications/storage_paper/Interspeech07-STD06-v13.pdf
14. Szoeke, I., Fapso, M., & Burget, L. (2008). Hybrid word-subword decoding for spken term detection. In *Proceedings of SIGIR*, Singapore (pp. 42-48): ACM.
15. Chen, S. F., & Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech & Language, 13*(4), 359–393.
16. Riedhammer, K., Do, V. H., & Hieronymus, J. (2013). A study on LVCSR and keyword search for tagalog. In *Proceedings of Interspeech*, (pp. 2529–2533). http://www.isca-speech.org/archive/interspeech_2013/i13_2529.html
17. Jeanrenaud, P., Eide, E., Chaudhari, U., McDonough, J., Ng, K., Siu, M., et al. (1995). Reducing word error rate on conversational speech from the Switchboard corpus. In *Proceedings of ICASSP*, (vol. 1, pp. 53–56): IEEE. doi:10.1109/ICASSP.1995.479271
18. BABEL Program. http://www.iarpa.gov/Programs/ia/Babel/babel.html.
19. Cui, J., Cui, X., Ramabhadran, B., Kim, J., Kingsbury, B., Mamou, J., et al. (2013). Developing speech recognition systems for corpus indexing under the IARPA babel program. In *Proceedings of ICASSP*, (pp. 6753–6757): IEEE. doi:10.1109/ICASSP.2013.6638969
20. Chen, N. F., Sivadas, S., Lim, B. P., Ngo, H. G., Xu, H., Pham, V. T., et al. (2014). Strategies for vietnamese keyword search. In *Proceedings of ICASSP*, (pp. 4149–4153). Florence: IEEE. doi:10.1109/ICASSP.2014.6854377
21. Metze, F., Rajput, N., Anguera, X., Davel, M., Gravier, G., Heerden, C. v., et al. (2012). The spoken web search task at mediaeval 2011. In *Proceedings of ICASSP*, (pp. 5165–5168). Kyoto: IEEE. doi:10.1109/ICASSP.2012.6289083
22. Metze, F., Anguera, X., Barnard, E., Davel, M., & Gravier, G. (2013). The spoken web search task at mediaeval 2012. In *Proceedings of ICASSP*, (pp. 8121–8125). Vancouver, BC: IEEE. doi:10.1109/ICASSP.2013.6639247
23. MediaEval Benchmarking Initiative for Multimedia Evaluation. http://www.multimediaeval.org/
24. Moseley, C. (Ed.). (2010). *Atlas of the world's languages in danger* (3rd ed.). Paris: UNESCO.
25. Tueske, Z., Nolden, D., Schlueter, R., & Ney, H. (2014). Multilingual MRASTA features for low-resource keyword search and speech recognition sysTEMS. In *Proceedings of ICASSP*, (pp. 7854–7858). Florence: IEEE. doi:10.1109/ICASSP.2014.6855129
26. Ghahremani, P., Babaali, B., Povey, D., Riedhammer, K., Trmal, J., & Khudanpur, S. (2014). A pitch extraction algorithm tuned for automaticspeech recognition. In *Proceedings of ICASSP*, (pp. 2494–2498). Florence: IEEE. doi:10.1109/ICASSP.2014.6854049
27. Lee, H.-Y., Zhang, Y., Chuangsuwanich, E., & Glass, J. (2014). Graph-based re-ranking using acoustic feature similarity between search results for spoken term detection on low-resource languages. In *Proceedings of Interspeech*, Singapore (pp. 2479–2483): ISCA. http://www.isca-speech.org/archive/interspeech_2014/i14_2479.html
28. Soto, V., Mangu, L., Rosenberg, A., & Hirschberg, J. (2014). A Comparison of multiple methods for rescoring keyword search lists for low resource languages. In *Proceedings of Interspeech*, Singapore (pp. 2464-2468). Singapore: ISCA. http://www.isca-speech.org/archive/interspeech_2014/i14_2464.html
29. Hartmann, W., Le, V.-B., Messaoudi, A., Lamel, L., & Gauvain, J.-L. (2014). Comparing decoding strategies for subword-based keyword spotting in low-resourced languages. In *Proceedings of Interspeech*, Singapore (pp. 2764–2768). http://www.isca-speech.org/archive/interspeech_2014/i14_2764.html
30. Hsiao, R., Ng, T., Zhang, L., Ranjan, S., Tsakalidis, S., Nguyen, L., et al. (2014). Improving semi-supervised deep neural network for keyword search in low resource languages. In *Proceedings of Interspeech*, Singapore (pp. 1088–1091): ISCA. http://www.isca-speech.org/archive/interspeech_2014/i14_1088.html
31. Cui, X., Kingsbury, B., Cui, J., Ramabhadran, B., Rosenberg, A., Rasooli, M. S., et al. (2014). Improving deep neural network acoustic modeling for audio corpus indexing under the IARPA babel program. In *Proceedings of Interspeech*, Singapore (pp. 2103–2107): ISCA. http://www.isca-speech.org/archive/interspeech_2014/i14_2103.html
32. Huang, J.-T., Li, J., Yu, D., Deng, L., & Gong, Y. (2013). Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *Proceedings of ICASSP*, Vancouver, BC (pp. 7304-7308): IEEE. doi:10.1109/ICASSP.2013.6639081
33. Chen, I.-F., Ni, C., Lim, B. P., Chen, N. F., & Lee, C.-H. (2014). A novel keyword+LVCSR-filler based grammar network representation for spoken keyword search. In *Proceedings of ISCSLP*, Singapore (pp. 192-196): IEEE. doi:10.1109/ISCSLP.2014.6936713
34. Chen, I.-F., Ni, C., Lim, B. P., Chen, N. F., & Lee, C.-H. (2015). A keyword-aware grammar framework for lvcsr-based spoken keyrowd search. In *Proceedings of ICASSP*, Brisbane: IEEE.
35. Sukkar, R. A., & Lee, C.-H. (1996). Vocabulary independent discriminative utterance verification for non-keyword rejection in subword based speech recognition. *IEEE Transactions on Speech and Audio Processing, 4*(6), 420–429.
36. Ou, J., Chen, K., Want, X., & Lee, Z. (2001). Utterance verification of short keywords using hybrid neural-network/HMM approach. In *Proceedings of ICII*, Beijing (vol. 2, pp. 671-676): IEEE. doi:10.1109/ICII.2001.983657.
37. Chen, I.-F., & Lee, C.-H. (2013). A hybrid HMM/DNN Approach to keyword spotting of short words. In *Proceedings of Interspeech*, Lyon (pp. 1574-1578): ISCA. http://www.isca-speech.org/archive/interspeech_2013/i13_1574.html
38. Chen, I.-F., & Lee, C.-H. (2013). A Resource-dependent approach to word modeling for keyword spotting. In *Proceedings of Interspeech*, Lyon (pp. 2544–2548): ISCA. http://www.isca-speech.org/archive/interspeech_2013/i13_2544.html
39. Szoke, I., Schwarz, P., Matejka, P., Burget, L., Karafiat, M., Fapso, M., et al. (2005). Comparison of Keyword spotting approaches for informal continuous speech. In *Proceedings of EuroSpeech*.
40. Mohri, M., Pereira, F., & Riley, M. (2008). Speech recognition with weighted finite-state transducers. In *Springer Handbook of Speech Processing* (pp. 559–584): Springer Berlin Heidelberg. doi:10.1007/978-3-540-49127-9_28
41. Allauzen, C., Mohri, M., & Roark, B. (2003). Generalized algorithms for constructing language models. In *Proceedings of ACL*, Stroudsburg, PA, USA (vol. 1, pp. 40–47): ACL. doi:10.3115/1075096.1075102
42. NIST Open Keyword Search 2013 Evaluation (OpenKWS13). http://www.nist.gov/itl/iad/mig/openkws13.cfm.
43. Povey, D., Ghoshal, A., Boulianne, G., L. S. B., Glembek, O. R., Goel, N., et al. (2011). The Kaldi speech recognition toolkit. In *Proceedings of IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, Hilton Waikoloa Village, Big Island, Hawaii, US: IEEE Signal Processing Society.
44. Vesely, K., Ghoshal, A., Burget, L., & Povey, D. (2013). Sequence-discriminative traning of deep neural networks. In *Proceedings of Interspeech*, Lyon, France (pp. 2345-2349): ISCA. http://www.isca-speech.org/archive/interspeech_2013/i13_2345.html
45. Novak, J. R., Minematsu, N., & Hirose, K. (2012). WFST-based Grapheme-to-Phoneme conversion: open source tools for alignment, model-building and decoding. In *Proceedings of International Workshop on Finite State Methods and Natural Language Processing*, Donostia-San Sebastian (pp. 45–49). https://code.google.com/p/phonetisaurus

46. Katz, S. M. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 35*(3), 400–401.



I-Fan Chen received his Master of Science degree from National Taiwan University in 2006. He is currently a Ph.D candidate at School of Electrical and Computer Engineering in the Georgia Institute of Technology. His Ph.D research is majorly focusing on resource-dependent acoustic and language modeling for spoken keyword search. Before joining Georgia Tech in 2010, Mr. Chen was a research assistant in the Institute of Information Science, Academia Sinica, Taiwan, during 2007 to 2009 and was involved in research works including attribute-based speech recognition and speaker recognition and diarization. His research interests include speech recognition, spoken keyword search, speaker recognition and diarization, and machine learning.

Chongjia Ni received the Ph.D. degree in engineering from National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2011. He is currently a scientist at Institute for Infocomm Research (I2R), A*STAR, Singapore. His research interests include automatic speech recognition, keyword search and machine learning.

B. P. Lim received his Ph.D. from the University of Illinois in Urbana-Champaign in 2011. He is currently a scientist at the Institute for Infocomm Research (I2R), Singapore. His research interests include acoustic modeling for speech recognition, speech processing for whispered speech, and spoken language processing for low-resource languages, keyword search, prototypes and applications. Dr. Lim is a member of the IEEE, and was active in the organizing committee for INTERSPEECH 2014. He is active in student mentoring for high-school science projects, some of which have led to accepted publications in IEEE conferences. He has received the MOE Outstanding Mentor Award multiple times for his mentorship.

Nancy F. Chen received her Ph.D. from the Massachusetts Institute of Technology (MIT) and Harvard University in 2011. She is currently a scientist at the Institute of Infocomm Research (I2R), Singapore. Her research interests include spoken language processing for low-resource languages, keyword search, spoken term detection, and computer-assisted language learning. Prior to joining I2R, she worked at MIT Lincoln Laboratory on her Ph.D. research, which integrates speech technology and speech science, with applications in speaker, accent, and dialect characterization. Dr. Chen is a reporter for the IEEE Speech and Language Processing Technical Committee Newsletter. She has also been active in organizing conferences such as Odyssey 2012: The Speaker and Language Recognition Workshop and INTERSPEECH 2014. Dr. Chen is a recipient of multiple awards, including the Microsoft-sponsored IEEE Spoken Language Processing Grant, the MOE Outstanding Mentor Award, and the NIH Ruth L. Kirschstein National Research Service Award.

Chin-Hui Lee is a professor at School of Electrical and Computer Engineering, Georgia Institute of Technology. Before joining academia in 2001, he had 20 years of industrial experience ending in Bell Laboratories, Murray Hill, New Jersey, as a Distinguished Member of Technical Staff and Director of the Dialogue Systems Research Department. Dr. Lee is a Fellow of the IEEE and a Fellow of ISCA. He has published over 400 papers and 30 patents, and was highly cited for his original contributions with an h-index of 66. He received numerous awards, including the Bell Labs President’s Gold Award in 1998. He won the SPS’s 2006 Technical Achievement Award for “Exceptional Contributions to the Field of Automatic Speech Recognition”. In 2012 he was invited by ICASSP to give a plenary talk on the future of speech recognition. In the same year he was awarded the ISCA Medal in scientific achievement for “pioneering and seminal contributions to the principles and practice of automatic speech and speaker recognition”.