



# Subword RNNLM Approximations for Out-Of-Vocabulary Keyword Search

Mittul Singh\*, Sami Virpioja†, Peter Smit\*‡, Mikko Kurimo\*

\*Department of Signal Processing and Acoustics, Aalto University, Espoo, Finland

†Department of Digital Humanities, Helsinki University, Helsinki, Finland

‡Inscripta, Helsinki, Finland

firstname.lastname@{aalto,helsinki}.fi

## Abstract

In spoken Keyword Search, the query may contain out-of-vocabulary (OOV) words not observed when training the speech recognition system. Using subword language models (LMs) in the first-pass recognition makes it possible to recognize the OOV words, but even the subword  $n$ -gram LMs suffer from data sparsity. Recurrent Neural Network (RNN) LMs alleviate the sparsity problems but are not suitable for first-pass recognition as such. One way to solve this is to approximate the RNNLMs by back-off  $n$ -gram models. In this paper, we propose to interpolate the conventional  $n$ -gram models and the RNNLM approximation for better OOV recognition. Furthermore, we develop a new RNNLM approximation method suitable for subword units: It produces variable-order  $n$ -grams to include long-span approximations and considers also  $n$ -grams that were not originally observed in the training corpus. To evaluate these models on OOVs, we setup Arabic and Finnish Keyword Search tasks concentrating only on OOV words. On these tasks, interpolating the baseline RNNLM approximation and a conventional LM outperforms the conventional LM in terms of the Maximum Term Weighted Value for single-character subwords. Moreover, replacing the baseline approximation with the proposed method achieves the best performance on both multi- and single-character subwords.

**Index Terms:** OOV, Keyword Search, Single character, RNNLM, first-pass

## 1. Introduction

The goal of Keyword Search (KWS) on audio data is to search for interesting terms (words or their sequences) in speech. These systems typically use an Automatic Speech Recognition (ASR) system in the background. The ASR system always fails to recognize words missing from its training data and thus finding interesting keywords containing these Out-of-Vocabulary (OOVs) words is a difficult task.

Projects funded to develop and improve Keyword Search, like the IARPA's BABEL program, have promoted a lot of work to improve the OOV Keyword Search [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. These studies have improved KWS by handling OOV keywords in mostly two ways. First, by replacing OOVs by their acoustically-similar proxies [1, 2, 3, 4] and second, by employing subword units instead of words to be able to recognize OOVs [5, 6, 7, 8, 9, 10, 11, 12, 13]. In the latter approach, subword-based conventional  $n$ -gram language models (LMs), with context sizes 2 to 4, have been successfully applied to first-pass recognition for OOV KWS. Here, using subword units is

This work was supported by the Academy of Finland (Flagship programme: Finnish Center for Artificial Intelligence, FCAI; Grants 320181, 320182, 320183).

Table 1: The table displays the length rates in #subwords per occurrence for frequent words (FW, training set frequency  $\geq 5$ ) on the training and test sets, and for OOV words on the test set from the Arabic and Finnish KWS described in Section 4. OOV Rate on these datasets is 2% approx. Length rates for both single- and multi-character subwords are presented.

Language	FW (Train)		FW (Test)		OOV (Test)	
	Single	Multi	Single	Multi	Single	Multi
Arabic	4.67	1.63	4.49	1.49	<b>6.80</b>	<b>3.32</b>
Finnish	7.03	1.08	7.25	1.07	<b>12.91</b>	<b>2.46</b>

crucial to model OOVs as with word-based LMs OOVs are lost and cannot be recovered in subsequent recognition passes.

Even with subword units, the conventional  $n$ -gram LM has limitations. They suffer from data sparsity issues leading to inaccurate scoring and hence, assigning low or failing to detect good hypotheses in the first or subsequent recognition passes [14]. In effect, causing KWS to not recognize OOV keywords. Using long-span neural network LMs, such as Recurrent Neural Networks (RNNs), can help with data sparsity, but they are prohibitively expensive to use in first-pass decoding [14]. Thus, researchers apply these models in the first-pass by approximating them to  $n$ -gram LMs [14, 15, 16, 17]. Inspired by these efforts, we develop a new method for RNNLM approximation to  $n$ -gram LMs for first-pass decoding in subword KWS.

In this paper, we also focus on capturing longer contexts in contrast to previous work on OOV KWS. This requirement is specially important when considering differently-sized subwords on morphologically-rich language datasets because OOV words are longer than frequent words, as shown in Table 1. Additionally, higher-order  $n$ -grams can be beneficial for capturing long-term dependencies in an approximated RNNLM. Hence, we introduce an  $n$ -gram-growing algorithm to our approximation method to facilitate building long-context  $n$ -gram LM versions of approximated RNNLMs (Section 3).

In the experiments, we train our baseline LMs and RNNLMs on single- and multi-character subwords. To evaluate these LMs on their OOV detection efficacy, we setup two KWS tasks on Arabic and Finnish datasets using only such OOV keywords that do not appear in the training data (Section 4 & 5).

## 2. Related Work

### 2.1. Approximating RNNs to Long-Span $n$ -gram LMs

There exist a few approximation techniques for converting RNNLMs to  $n$ -gram LMs: variational approximation [14], probability-conversion [15] and iterative conversion [15, 16]. Prior work [15] compared these techniques and the best approximating technique – iterative conversion – outperformed other methods using smaller order  $n$ -grams in a speech recognition task. However, iterative conversion method's effective-

ness on low-order word  $n$ -grams seems sub-optimal for OOVs. The OOVs are usually longer than frequent words (refer Table 1) and when represented as subwords (like single- and multi-characters) require LMs which can capture long-term information well. Among other methods, the variational approximation method is also not a good fit for our purposes, as it can fail to sample rare subword contexts, which might be present in the subword sequences of OOVs. Hence, we use the probability-conversion method to develop a more efficient method and introduce an  $n$ -gram-growing variant of the algorithm for approximating subword RNNLMs to larger context sizes.

## 2.2. Subword-based Keyword Search for OOV Prediction

Subword-based Keyword search for OOVs has been performed using either phonetic units like phones, graphemes, syllables and sequences of phones [5, 7, 8, 9, 10, 11, 12] or textual single- or multi-character units [6, 13].

The textual approach saves effort spent on generating phonetic representations of OOV words, which has been a common trend across the phonetic approach to KWS systems. Our KWS system is similar to the second category of subword-based KWS systems where we use single- and multi-character units for decoding. Although multi-character subword units have been used for decoding before, single-character units for decoding have been found to obtain poor performance in a KWS system [6]. However, this may be an effect of an ASR system that does not work well on character units. For the datasets considered in this work, we apply tools that allow applying grapheme-based acoustic models for subword ASR improving the performance for character-based ASR compared to word-based ASR [18].

## 3. Approximating RNNLMs to $n$ -grams

Given a corpus of subword units ( $w_i$ ) and a corresponding vocabulary ( $V$ ), RNNs ( $p_r$ ) can be approximated to  $n$ -grams ( $p_n$ ) using the probability-conversion (PC) method [15]. For an  $n$ -gram history of subwords  $h$  and its back-off history  $\bar{h}$ , PC marginalizes over observed sentence histories ( $b$ ) that precede  $h$  in the RNN scoring function  $y(w|h)$ .  $y(w|h)$  is then used in a back-off LM ( $p_n^{PC}(w|h)$ ) to approximate the RNN:

$$y(w|h) = \sum_{b \in H_{*|h}w} \frac{c(bhw)}{c(hw)} \cdot p_r(w|bh)$$

$$p_n^{PC}(w|h) = S \cdot \frac{y(w|h)}{\sum_{v \in V} y(v|h)} + (1 - S) \cdot p_n^{PC}(w|\bar{h}) \quad (1)$$

Here  $H_{*|h}$  is the set of observed sentence histories preceding  $h$ ,  $c(x)$  is the count of a sequence  $x$ , and  $0 < S < 1$  is the smoothing factor.

As subword LMs for OOVs will need higher-order  $n$ -grams, the above method can be inefficient due to the normalization term in (1) and can only produce  $n$ -gram LMs of order three to five effectively. Subverting the normalization calculation, we can also consider the marginalization on sentence histories ( $b$ ) in a novel way for approximating RNNs, whereby,

$$p_n^{\text{Ours}}(w|h) = \sum_{b \in H_{*|h}} p(w, b|h)$$

$$= \sum_{b \in H_{*|h}} p(b|h)p(w|bh)$$

$$\approx \sum_{b \in H_{*|h}} \frac{c(bh)}{c(h)} \cdot p_r(w|bh) \quad (2)$$

The above formula provides a proper distribution with the requirement that RNNLM output vector ( $O(bh)$ ) for every context  $bh$  in the corpus is available. For a large corpus ( $C$ ) and vocabulary ( $V$ ), storing and using the complete output vector for every context becomes as resource intensive as (1), requiring  $\mathcal{O}(|C| \cdot |V|)$  storage capacity and operations.

To reduce the resource requirements, we only store the probabilities  $p_r(w|bh)$  for the context  $bh$  under two conditions: the  $n$ -gram is observed in the corpus ( $c(bhw) > 0$ ), or  $w$  is included in  $O_u^{\text{top}K}(bh)$ , the top  $K$  probabilities of the RNN's output vector for  $bh$  where the next word  $u \neq w$ . This restriction considers only a part of the output vector instead of the complete output vector, and consequently, lowers the complexity to  $\mathcal{O}(|C| \cdot K + 1)$ . We implement the restriction by considering the contexts  $C^{1+\text{top}K}$  that fulfill one of the above two conditions:  $C^{1+\text{top}K} = \{bhw \mid c(bhw) > 0 \vee (\exists u \in V : c(bhu) > 0 \wedge w \in O_u^{\text{top}K}(bh) \wedge w \neq u)\}$ . Hence, (2) changes to:

$$p_n^{\text{Ours}}(w|h) = \sum_{b \in H_{*|h}} \frac{c(bh)}{c(h)} \cdot \mathbb{1}_{bhw \in C^{1+\text{top}K}} \cdot p_r(w|bh) \quad (3)$$

where  $\mathbb{1}_{bhw \in C^{1+\text{top}K}}$  is the indicator function to represent contexts from the set  $C^{1+\text{top}K}$ . In both (2) and (3), calculating  $c(bh)$  is not necessary and we can also just calculate the sum of probabilities for an  $n$ -gram  $bhw$ .

The formulation in (3) is not a proper distribution, and using small values of  $K$  (e.g.  $K = 0, 1$  etc.), the missing probability mass can be high when creating backing-off LMs. Still, for small values of  $K$ , it provides a further speed up in comparison to (2) for approximating RNNLMs.

Still creating an approximated RNNLM for higher-order  $n$ -grams using (3) can be prohibitively expensive. Hence, we embed (3) in an  $n$ -gram-growing algorithm [19]. This algorithm<sup>1</sup> can iteratively grow the  $n$ -grams in the LM, selecting important  $n$ -gram contexts using a cost function based on minimum description length of the model and the data. In this algorithm, we can also specify two parameters to limit the number of  $n$ -grams: the minimum threshold for accepting an  $n$ -gram, and maximum context size ( $n$ ). For brevity, we do not describe details of the growing algorithm here, but refer readers to [19].

## 4. Experimental Setup

We setup the experiments on publicly-available datasets from two languages: Arabic and Finnish. Both languages are morphologically rich, leading to quite a few OOV words in the datasets (OOV rate  $\sim 2\%$ ).

### 4.1. Datasets

For Arabic acoustic models, we used the training corpus from the MGB-2 challenge [20], consisting of 1,200 hours of Aljazeera's television programs data. For testing, we used the MGB-2 development set, which has eight hours of data and 57k words. For Arabic language models, we used a corpus of 130 million tokens obtained from the Aljazeera's website. This text contains around 1.4 million unique words. For Finnish acoustic models, we used 1500 hours of Finnish audio data from three different data sets, namely, the Speecon corpus [21], the Speechdat database [22] and the parliament corpus [23]. For testing, we used a set of broadcast news from the Finnish national broadcaster (Yle) containing 5 hours of speech and 35k

<sup>1</sup>The algorithm (3) implementation is available at <https://github.com/lallubharteja/varikn>

Table 2: KWS performance for both morphs and single characters is presented on Arabic and Finnish datasets along with language model size in number of  $n$ -grams. The table reports MTWV and Lattice Recall calculated on first-pass decoded lattices for KNV, RNN5, RNNV and its linear interpolation (KNV+RNNV) with equal weights.

Segmentation	LM	Arabic			Finnish		
		MTWV	Lattice Recall	#N-grams	MTWV	Lattice Recall	#N-grams
morphs	KNV	0.551	0.505	6.13M	0.317	0.299	4.18M
	RNN5 (PC)	0.245	0.244	279K	0.128	0.116	1.24M
	KNV+RNN5 (PC)	0.560	0.531	6.27M	0.308	0.289	4.85M
	RNNV (Ours)	0.328	0.350	109K	0.202	0.183	1.46M
	KNV+RNNV (Ours)	<b>0.591</b>	<b>0.564</b>	6.19M	<b>0.321</b>	<b>0.303</b>	4.71M
characters	KNV	0.561	0.551	5.56M	0.686	0.650	6.75M
	RNN5 (PC)	0.595	0.610	501K	0.704	0.693	636K
	KNV+RNN5 (PC)	0.646	0.634	5.87M	0.715	0.696	7.04M
	RNNV (Ours)	0.659	<b>0.689</b>	618K	0.699	0.687	548K
	KNV+RNNV (Ours)	<b>0.692</b>	0.673	5.90M	<b>0.738</b>	<b>0.721</b>	6.94M

words [23]. For Finnish, we train the language models on the Finnish Text Collection [24]. The training set consists of 143M tokens with 4.2M unique types.

## 4.2. Keyword Search Setup

The subword KWS requires a subword-based ASR system. The ASR system is set up using the Kaldi toolkit [25] in a similar fashion to the subword systems presented in [26]. These systems apply grapheme-based acoustic models that can generate pronunciations for all subwords. This setup also requires modification of the weighted finite-state transducer of the lexicon (L-FST) for treating subword units similarly to words in Kaldi. For the details of the modification, see [18].

For setting up the subword keyword search task, we create a list of keywords from the language-specific test sets. We extract the OOV words from the evaluation set, but remove any OOVs which have only a single character or have a character that is not present in the words of the training set; and for Finnish are spelled incorrectly. Thus, we obtain 449 and 661 OOV keywords for the Arabic and Finnish test sets, respectively.<sup>2</sup>

Next, we segment these keywords to single- and multi-character subwords; and apply scripts from Kaldi’s openKWS system [27] to setup the KWS task for the different segmentations. To evaluate our models in keyword search, we use the Maximum Term Weighted Value (MTWV) on the first-pass lattices. MTWV is a popular KWS metric which incorporates both keyword-specific misses and false alarms. For further details on MTWV, see [28]. We also measure each lattice’s keyword recall, which measures the amount of correctly retrieved keywords out of the relevant ones.

## 4.3. Building Language Models

As subwords, we use single **characters** and the multi-character units created by Morfessor Baseline [29, 30]. For brevity, we refer to the latter units as **morphs**, although they do not correspond to linguistic morphemes. For Arabic, we mark the right end of subwords except when at the end of a word. E.g. (international = inter+ nation+ al). For Finnish, we mark both the left and right ends of subwords except when at the beginning or the end of a word. E.g. (international = inter+ +nation+ +al). These choices are based on prior work [26], which shows that the respective marking schemes for Arabic and Finnish datasets outperform other schemes.

<sup>2</sup>We publish the OOV lists for these tasks at <https://github.com/lallubharteja/KWS-Scripts>

As the  $n$ -gram LM baselines, we train the Kneser-Ney [31] smoothed variable-length  $n$ -grams (**KNV**) using the VariKN toolkit [19]. For first-pass decoding, the RNNLMs, built similarly to the *small* architectures from [26], are approximated using RNN approximation method from Section 3 with  $K = 3$  and denoted as **RNNV**. For RNNV LMs, we use a threshold of 0.1 for both types of subword segmentations. For Arabic, the best **RNNV** is obtained for  $n$ ’s 13 and 6 for morphs and characters respectively. For the Finnish,  $n$ ’s 11 and 5 prove best for morphs and character **RNNV** models, respectively.

For comparison with our method, we also build LMs using the Probability-Conversion method (PC). Creating higher-order LMs can be expensive with the PC method, so we approximate RNNLMs to 5-gram models (**RNN5**) using this method. The **RNN5** are constrained to have similar number of 5-grams as in the corresponding **RNNV** to keep the model strength of 5-gram LMs comparable. On the Arabic and Finnish text, creating **RNN5** is 2.3/12 and 1.6/6.7 times slower than building the corresponding **RNNV** with characters/morphs respectively.

Additionally, we linearly interpolate **RNNV** and **RNN5** LMs with **KNV** using equal weights, creating **KNV+RNNV** and **KNV+RNN5**. We note that the RNNLM approximations and their interpolated versions applied to the first-pass had a worse ASR performance than applying **KNV** for the same, but are not presented here for conciseness. The perplexities of these models cannot be compared, as the approximated RNNs are not proper distributions.

## 5. Keyword Search Experiments

In this work, we perform KWS experiments for Arabic and Finnish OOV words from the evaluation set, which are the hardest to predict as they are unseen in the training set. The word-based LMs that are devoid of any phonetic information will fail at predicting them and have a zero MTWV. Hence, we setup a subword-based keyword search on the above datasets.

Table 2 compares the different LMs from Section 4.3 on the Arabic and Finnish OOV Keyword Search. These LMs are built using morph and single-character subwords. The results in Table 2 show that single-character models perform better than the morph-based models mostly because OOVs can be better represented at character level than with morphs. Similarly, the RNNLM approximations (**RNN5** and **RNNV**) have a more competitive performance with **KNV** at character-level than with the morph units. Furthermore, both the interpolated models (**KNV+RNN5** and **KNV+RNNV**) outperform **KNV** on characters with larger improvements (at least 15.1% on Arabic

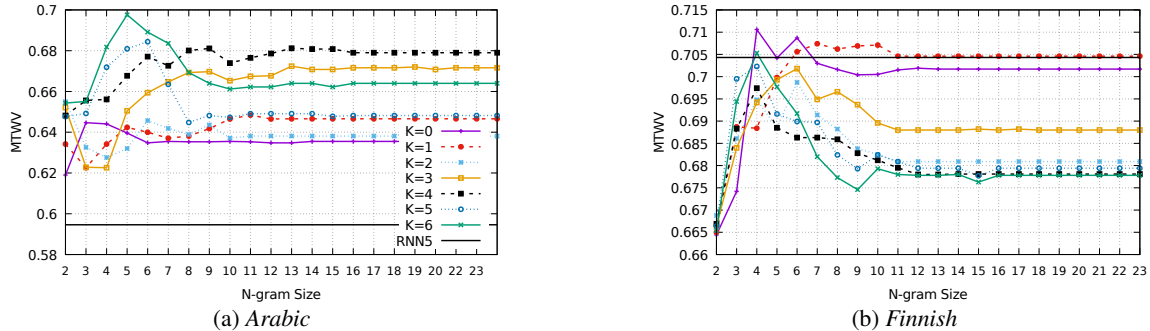


Figure 1: Single-character **RNNV**'s MTWV measured by varying the  $K$  values in the top- $K$  step of the algorithm (3) for Arabic and Finnish KWS tasks. **RNN5** as constructed in Section 4.3 is shown by the thick horizontal line.

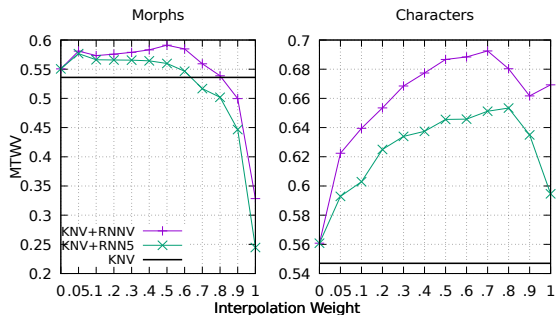


Figure 2: Interpolated LMs' MTWV measured by varying the interpolation weight of RNNLMs approximations in in **KNV+RNN5** and **KNV+RNNV** models for Arabic KWS task. **KNV** MTWV is displayed for reference.

and 4.2% on Finnish) than on morphs, where the improvements at best can be 7.2% on Arabic and 1.2% on Finnish.

On Arabic and Finnish KWS, **KNV+RNNV** achieves larger improvements over **KNV** for different subwords on the Arabic KWS (7.2%–23.3%) than on the Finnish KWS (1.2%–7.6%). Improved **KNV** performance on Finnish might be dependent on the quality of the underlying acoustic models. In particular, the Finnish ASR has a larger and cleaner (Speecon and Speechdat transcripts are verified) dataset than used in the Arabic ASR. Also, Finnish has a simpler phonetic structure than Arabic.

Across the different subword units, **RNNV** mostly performs better than **RNN5** on MTWV, except when using characters on the Finnish KWS. The performance differences are larger across Arabic and Finnish when comparing the interpolated models, with **KNV+RNNV** performing the overall best. This shows the benefits of using the proposed method, which has a different scoring scheme and access to higher-order  $n$ -gram contexts than the probability-conversion method.

## 6. Analysis

In this section, we analyse KWS performance's sensitivity with respect to important parameters involved in construction of **RNNV** and the interpolated models. For **RNNV**, we look at the choice of  $K$  in the top- $K$  step of the algorithm (3) and the  $n$ -gram context size. For the interpolated models, we consider the interpolation weight of the RNNLM approximations in **KNV+RNN5** and **KNV+RNNV** models.

### 6.1. $N$ -gram size & top- $K$ values in RNNLM approximation

In Figure 1, we report KWS performance when varying the  $K$  from 1 to 6 and  $n$  from 5 to 23 for the character models while

keeping the growing algorithm's threshold fixed at 0.1. A fixed threshold forces an  $(n+1)$ -gram LM to have the same  $n$ -grams as the  $n$ -gram LM had before growing to the  $(n+1)$ -grams.

On both Arabic and Finnish, we find **RNNV** for  $K = 0$  can outperform **RNN5**. Lowering the  $K$  allows faster construction of **RNNV** and thus, improves benefits over PC-based method. Overall, the performance for all  $K$ s seem to vary more with  $n < 4$  and then stabilize with  $n \geq 4$  for some  $K$ s even outperforming the smaller  $n$ . These improvements suggest that the system can benefit from longer contexts  $n \geq 4$ . These observations suggest that choosing  $K$  and  $n$  are important to enable efficient and improved performance of **RNNV**.

### 6.2. Varying the RNNLM Interpolation Weight

Figure 2 show the variation of MTWV against different interpolation weights (in the range of  $[0, 1]$ ) of **RNN5** and **RNNV** in their respective interpolated models for the Arabic KWS task. For some interpolation weights, **KNV+RNN5** and **KNV+RNNV** performance can be further improved than results observed in Table 2. In most cases, **KNV+RNNV** achieves a better performance than **KNV+RNN5** suggesting that **RNNV** is able to complement **KNV** better than **RNN5**. Similar trends were also observed for the Finnish KWS task.

## 7. Concluding Remarks

In this paper, we introduced a new efficient technique inspired from probability-conversion method to approximate RNNLMs to  $n$ -gram LMs. We also extended this technique using  $n$ -gram-growing algorithm to better handle OOVs and create better long-span approximations of RNNLM.

Using multi- and single-character subwords, we constructed interpolated LMs using conventional  $n$ -gram and approximated RNN models. We applied these models on first-pass-based Arabic and Finnish Keyword Search for OOVs, which are the hardest to predict. We observed that our method, which had longer contexts, complemented the conventional  $n$ -gram LMs better than the probability-conversion method. In addition, single-character-based LMs outperformed the morph-based LMs and using the proposed method single-character models performed the best overall. In future, we would also like to investigate the effect of rescoring on this KWS setup.

While predicting OOVs in a high-resource scenario, we were able to achieve MTWVs higher than IARPA's BABEL Program aim of 0.5 MTWV. Still, we want to explore if similar performance can be obtained on an under-resourced scenario as prescribed by IARPA's BABEL program.

## 8. References

- [1] B. Logan and J. V. Thong, "Confusion-based query expansion for OOV words in spoken document retrieval," in *ICSLP*, 2002.
- [2] G. Chen, O. Yilmaz, J. Trmal, D. Povey, and S. Khudanpur, "Using proxies for oov keywords in the keyword search task," in *ASRU*, 2013, pp. 416–421.
- [3] L. Mangu, B. Kingsbury, H. Soltau, H. Kuo, and M. Picheny, "Efficient spoken term detection using confusion networks," in *ICASSP*, 2014, pp. 7844–7848.
- [4] B. Gundogdu, B. Yusuf, and M. Saraclar, "Generative rnns for oov keyword search," *IEEE Signal Processing Letters*, vol. 26, no. 1, pp. 124–128, 2019.
- [5] I. Szöke, L. Burget, J. Cernocký, and M. Fapso, "Sub-word modeling of out of vocabulary words in spoken term detection," in *SLT*, 2008, pp. 273–276.
- [6] W. Hartmann, V. Bac Le, A. Messaoudi, L. Lamel, and J. Gauvain, "Comparing decoding strategies for subword-based keyword spotting in low-resourced languages," in *INTERSPEECH*, 2014, pp. 2764–2768.
- [7] I. Bulyko, J. Herrero, C. Mihelich, and O. Kimball, "Subword speech recognition for detection of unseen words," in *INTERSPEECH*, 2012.
- [8] D. Karakos and R. Schwartz, "Subword and phonetic search for detecting out-of-vocabulary keywords," in *INTERSPEECH*, 2014.
- [9] Y. He, P. Baumann, H. Fang, B. Hutchinson, A. Jaech, M. Ostendorf, E. Fosler-Lussier, and J. B. Pierrehumbert, "Using pronunciation-based morphological subword units to improve OOV handling in keyword search," *IEEE/ACM Transactions on Audio, Speech & Language Processing*, vol. 24, no. 1, pp. 79–92, 2016.
- [10] P. Yu and F. Seide, "A hybrid word / phoneme-based approach for improved vocabulary-independent search in spontaneous speech," in *ICSLP*, 2004.
- [11] F. Seide, P. Yu, C. Ma, and E. Chang, "Vocabulary-independent search in spontaneous speech," in *ICASSP*, 2004, pp. 253–256.
- [12] S. Lee, K. Tanaka, and Y. Itoh, "Generating complementary acoustic model spaces in dnn-based sequence-to-frame DTW scheme for out-of-vocabulary spoken term detection," in *INTERSPEECH*, 2016, pp. 755–759.
- [13] Y. Khokhlov, I. Medennikov, A. Romanenko, V. Mendelev, M. Korenevsky, A. Prudnikov, N. Tomashenko, and A. Zatornitsky, "The stc keyword search system for openkws 2016 evaluation," in *INTERSPEECH*, 2017, pp. 3602–3606.
- [14] A. Deoras, T. Mikolov, S. Kombrink, M. Karafit, and S. Khudanpur, "Variational approximation of long-span language models for LVCSR," in *ICASSP*, May 2011, pp. 5532–5535.
- [15] H. Adel, K. Kirchhoff, N. T. Vu, D. Telaar, and T. Schultz, "Comparing approaches to convert recurrent neural networks into back-off language models for efficient decoding," in *INTERSPEECH*, 2014, pp. 651–655.
- [16] E. Arisoy, S. F. Chen, B. Ramabhadran, and A. Sethy, "Converting neural network language models into back-off language models for efficient decoding in automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech and Lang. Proc.*, vol. 22, no. 1, pp. 184–192, Jan. 2014.
- [17] M. Suzuki, N. Itoh, T. Nagano, G. Kurata, and S. Thomas, "Improvements to n-gram language model using text generated from neural language model," in *ICASSP 2019*, May 2019, pp. 7245–7249.
- [18] P. Smit, S. Virpioja, and M. Kurimo, "Improved subword modeling for wfst-based speech recognition," in *INTERSPEECH*, 2017, pp. 2551–2555.
- [19] V. Siivola, T. Hirsimäki, and S. Virpioja, "On growing and pruning kneser-ney smoothed n-gram models," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1617–1624, July 2007.
- [20] A. M. Ali, P. Bell, J. R. Glass, Y. Messaoui, H. Mubarak, S. Renals, and Y. Zhang, "The MGB-2 challenge: Arabic multi-dialect broadcast media recognition," in *SLT*, 2016, pp. 279–284.
- [21] D. J. Iskra, B. Grosskopf, K. Marasek, H. van den Heuvel, F. Diehl, and A. Kieling, "Speecon - speech databases for consumer devices: Database specification and validation." in *LREC*. European Language Resources Association, 2002.
- [22] A. Rosti, A. Rämö, T. Saarelainen, and J. Yli-Hietanen, "Speechdat Finnish database for the fixed telephone network," Tampere University of Technology, Tech. Rep., 1998.
- [23] A. Mansikkaniemi, P. Smit, and M. Kurimo, "Automatic construction of the Finnish parliament speech corpus," in *INTERSPEECH*, 2017-08, pp. 3762–3766.
- [24] "The Helsinki Korp version of the Finnish text collection," <http://urn.fi/urn:nbn:fi:lb-201403268>, accessed: 2018-06-29.
- [25] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *ASRU*. IEEE Signal Processing Society, Dec. 2011.
- [26] P. Smit, S. R. Gangireddy, S. Enarvi, S. Virpioja, and M. Kurimo, "Character-based units for unlimited vocabulary continuous speech recognition," in *ASRU*, 2017, pp. 149–156.
- [27] J. Trmal, M. Wiesner, V. Peddinti, X. Zhang, P. Ghahremani, Y. Wang, V. Manohar, H. Xu, D. Povey, and S. Khudanpur, "The kaldi openkws system: Improving low resource keyword search," in *INTERSPEECH*, 2017, pp. 3597–3601.
- [28] S. Wegmann, A. Faria, A. Janin, K. Riedhammer, and N. Morgan, "The tao of atwv: Probing the mysteries of keyword search performance," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 192–197.
- [29] M. Creutz and K. Lagus, "Unsupervised discovery of morphemes," in *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning - Volume 6*, ser. MPL '02, 2002, pp. 21–30.
- [30] S. Virpioja, P. Smit, S. Grnroos, and M. Kurimo, "Morfessor 2.0: Python implementation and extensions for morfessor baseline," Aalto University; Aalto-yliopisto, Tech. Rep., 2013. [Online]. Available: <http://urn.fi/URN:ISBN:978-952-60-5501-5>
- [31] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *ICASSP*, vol. 1, 1995, pp. 181–184 vol.1.