# CROSS-WORD SUB-WORD UNITS FOR LOW-RESOURCE KEYWORD SPOTTING

*William Hartmann, Lori Lamel, and Jean-Luc Gauvain*

Spoken Language Processing Group, LIMSI-CNRS
91403 Orsay, France
{hartmann, lamel, gauvain}@limsi.fr

## ABSTRACT

We investigate the use of sub-word lexical units for the detection of out-of-vocabulary (OOV) keywords in the keyword spotting task. Sub-word units based on morphological decomposition and character ngrams are compared. In particular, we examine the benefit of sub-word units that cross word boundaries. Experiments are performed on the IARPA Babel Turkish dataset. Our results demonstrate that cross-word sub-word units achieve similar performance on OOV keywords as other types of sub-word units, but can be combined to produce further gains. We also show that sub-word units can be used to improve detection of in-vocabulary keywords. System combination provides a 18% relative gain in ATWV with the best two systems, and 25% with the best three systems.

***Index Terms—*** keyword search, spoken term detection, OOV, sub-word lexical units, low resource LVCSR

## 1. INTRODUCTION

Recently there has been an increased interest in the task of keyword spotting (KWS)—this task is also referred to as Spoken Term Detection (STD) in the literature. The task differs significantly from the more traditional speech transcription task; performance measures for one task are not necessarily predictive for the other. Instead of accurately transcribing the entire utterance, a system only needs to determine whether a specified set of keywords are present.

Several approaches to KWS have been proposed over the years. Some of the earliest systems were template-based approaches using dynamic time warping (DTW) [1]. These were followed by HMM-based systems that used a model for the keyword and one or more models for all non-keywords [2]. More recent work utilizes a two-stage approach that leverages traditional ASR systems [3, 4]. The speech is initially decoded with an ASR system, producing either a 1-best transcript, n-best list, or lattice. Keyword search is then performed on the output of the ASR system. As with other recent work [5, 6], we adopt this two-stage approach and search the resulting lattices.

One particular difficulty for KWS is handling out-of-vocabulary (OOV) keywords, especially in the low-resource setting. While OOV words are typically a low-frequency occurrence—with a significant, but limited effect on word error rate (WER)—they can comprise a significant portion of the keywords in any KWS task [7]. Many approaches have previously been proposed to address the OOV issue. In [8], word lattices are converted to phone lattices; keywords are converted to phone strings and searched in the phone lattice. This can be extended to converting the word lattice to a sub-word lattice [9], or decoding with sub-word units [10]. An alternative is to expand the keywords by searching for in-vocabulary (IV) words that are easily confusable with OOV words [11]

In this work, we also focus on the use of sub-word units. In addition to the previously seen morphological decompositions [12] and character ngrams [13], we explore the use of cross-word sub-word units—sub-word units that can span word boundaries. We note that Bulyko et al. [14] also included a system that incorporated cross-word units, but it is unclear how those units affected performance—particularly since all keywords were single words. Our hypothesis is that this will allow OOV words to be discovered without relying on single character units. While the majority of the prior work focuses on OOV words, we also demonstrate the potential for improving performance on (IV) keywords.

We detail the methods used to generate sub-word units in Section 2. The dataset and keyword spotting system are described in Section 3. Section 4 contains general results and Section 5 provides a more detailed analysis. Conclusions are presented in Section 6.

## 2. SUB-WORD LEXICAL UNITS

We propose to use sub-word models to handle the detection of OOV keywords. While our focus is evaluating the efficacy of cross-word sub-word models, we will also describe a standard morphological decomposition approach. We use Morfessor [15] for performing morphological decomposition. Morfessor computes a generative probabilistic model given a list of words and the count of their occurrences. The model learns a set of morphological units that can be combined to represent any word in the corpus. These units represent a trade-off between maximizing the likelihood of the data and minimizing

| Lexical Units | Total | Average Length |
|---|---|---|
| Baseline | 10110 | 4.7 |
| Morfessor | 6187 | 4.0 |
| 3gram-wi | 4071 | 2.4 |
| 5gram-wi | 6624 | 3.4 |
| 7gram-wi | 6093 | 3.6 |
| 3gram-cw | 6855 | 2.6 |
| 5gram-cw | 7561 | 3.4 |
| 7gram-cw | 7302 | 3.6 |

**Table 1**. Number of unique units in the lexicon and average length of those units. Note the average length is computed on the training transcript.

the total number of units.

Our approach for building the remaining sub-word units relies on character ngrams, similar to [13]. The frequency of every possible character-level ngram (for a specific $n$) is computed over the training corpus. In order to limit the eventual size of the lexicon, we only keep the most frequent ngrams—in this work we keep the 15k most frequent, but preliminary experiments showed the number of initial units had little effect on the final lexicon. Given the initial set of sub-word units, the training corpus is segmented to minimize the total number of words in the segmented transcript. This heuristic for segmentation is slightly different than the greedy "longest match" approach [16], but is equivalent to segmenting with a uniform language model.

The segmentation is further improved through an iterative process. A trigram language model is built from the segmented corpus. Using the language model, the original corpus is resegmented. This process is repeated until convergence—approximately 3 to 5 iterations for our experiments.

Since we are considering both word-internal and cross-word sub-word units, the space between words must be handled. We do not treat white space as a separate character, but instead attach it to the final character of a word. Therefore the white space does not count as a character when we compute the character ngrams. For the word-internal units, the character representing white space can only be present in the final character of the unit, while it can appear anywhere in a cross-word unit. Maintaining a representation of white space in the lexical units provides the additional benefit of allowing for a unique conversion of the recognizer output in terms of sub-word units to words.

For both types of sub-word units, we build three sets of units. In each case we consider character ngrams of order 3, 5, and 7. The word-internal sub-word units are referred to as *3gram-wi*, *5gram-wi*, and *7gram-wi* throughout the remainder of the paper. Cross-word sub-word units are referred to as *3gram-cw*, *5gram-cw*, and *7gram-cw*. Details regarding the size of the lexicon and the average length of the lexical units are in Table 1. The lexicons for the cross-word units are larger than their counterparts, but the average lengths are similar.

## 3. EXPERIMENTAL SETUP

### 3.1. IARPA Babel Data

Our experiments are performed on the IARPA Babel Turkish data, specifically the "IARPA-babel105b-v0.4-sub-train" dataset. The particular release contains only 10 hours of transcribed conversational telephone speech for training. While a pronunciation lexicon is provided with the data, we do not use it. In keeping with the low-resource approach, we do not assume any *a priori* knowledge of the acoustic units or pronunciation lexicon. Results are reported on the 10 hour development set. A two hour subset of the development set was used for tuning parameters in the final KWS system. This setup is similar to the Tagalog system used in [11].

The keyword list contains both single and multi-word keywords. The full list contains 3291 keywords, though, only 1778 keywords are present in the development data. Many of the keywords not in the development set are found in the evaluation data. Since the scoring procedure does not consider keywords with zero occurrences, our keyword list effectively contains just those 1778 keywords. Of those keywords appearing in the development set, 421 contain at least one word not seen during training—an OOV rate of 24%. In general, the keywords are rare events. Over the 10 hour development set, each keyword appears an average of 5 times; 30% of the keywords only appear once.

### 3.2. ATWV Metric

The performance metric defined for this task is the "actual term weighted value" (ATWV), and follows the definition used in the NIST 2006 Spoken Term Detection evaluation [17]. The total ATWV is the mean of the ATWV scores for each individual keyword. The keyword specific ATWV for keyword $k$ can be computed by

$$\text{ATWV}(k) = 1 - P_{FR}(k) - \beta P_{FA}(k) \qquad (1)$$

where $P_{FR}$ and $P_{FA}$ refer to the probability of a false reject (miss) and false accept, respectively. A trade off between false rejects and false accepts is maintained by the constant $\beta$. According to the NIST protocol, we set the value of $\beta$ to 999.9.

It is important to recognize that the ATWV metric is not dependent on the frequency of any specific keyword. A keyword that appears once and a keyword that appears hundreds of times will contribute equally to the final score. This also highlights why detection of OOV keywords is such an important factor for KWS performance.

### 3.3. KWS System

We use the Kaldi speech recognition toolkit [18] for all of our experiments. As mentioned previously, we do not assume a pronunciation lexicon, so we use a grapheme-based lexicon instead—our preliminary experiments did not find a significant performance difference between a phone-based and a grapheme-based lexicon in terms of WER. The use of a grapheme-based lexicon also simplifies the process of generating pronunciations for sub-word units.

Standard PLP features with a pitch estimation feature were used to build an initial model. These features were then transformed by LDA using a context window of 9 frames as input; the features were projected down to 40 dimensions. The LDA transformation is also combined with MLLT and speaker-dependent fMLLR during training. The final model uses subspace Gaussian mixture models (SGMM) for the states [19], discriminatively trained using boosted maximum mutual information (BMMI) [20]. A trigram language model is estimated from the training transcripts using Kneser-Ney smoothing [21].

Word-level lattices are generated through multiple decoding passes. The first two passes use the standard GMM system. The lattices are finally rescored using the SGMM+BMMI models. The word-level lattices are converted to indices as proposed in [22]. Keywords are compiled into acceptors. For sub-word models, the keyword acceptors accept any sequence of sub-word units that can be combined to make the keyword, not just a single decomposition. Detection hypotheses are obtained by composing the acceptors with the indices. Before evaluating the results, a detection threshold must be set. We use the method proposed in [23] that uses a keyword-specific threshold based on expected keyword counts. Scoring is performed by the NIST F4DE tool.

## 4. RESULTS

### 4.1. Single System

ATWV results for each individual system are shown in Table 2. If we consider only the performance on all keywords, only the Morfessor-derived units appear to offer any benefit over the baseline system. Once we examine the performance on IV and OOV keywords separately, the picture changes. Each system detects a significant portion of the OOV keywords. The only reason the Morfessor result improves over the baseline on all keywords is that it detects some OOV keywords without reducing IV performance. The other sub-word systems all see a large decrease in IV performance, resulting in a lower overall score, even though they all outperform the Morfessor units on the OOV keywords.

If we focus solely on OOV performance, the two 3gram sub-word systems perform best. The incorporation of cross-word units does not appear to affect the performance much—though they do have worse IV performance. It is unclear why the *7gram* systems outperform the *5gram* systems in terms of OOV, but it might be related to the relative size of their lexicons (see Table 1). Assuming we are only using the sub-word models for OOV keywords, the best performing system would simply combine the baseline IV results and either of the *3gram* OOV results. We will now examine the combination of the systems in more detail.

| Lexical Units | All | IV | OOV |
|---|---|---|---|
| Baseline | 0.2186 | 0.2864 | 0.0000 |
| Morfessor | 0.2387 | 0.2870 | 0.0830 |
| 3gram-wi | 0.2212 | 0.2538 | 0.1160 |
| 5gram-wi | 0.2120 | 0.2481 | 0.0958 |
| 7gram-wi | 0.2073 | 0.2376 | 0.1094 |
| 3gram-cw | 0.2087 | 0.2372 | 0.1168 |
| 5gram-cw | 0.2055 | 0.2403 | 0.0932 |
| 7gram-cw | 0.1966 | 0.2281 | 0.0952 |

**Table 2**. ATWV results for the baseline word-based system and the sub-word systems described in Section 2. The *All* column shows results over all keywords while *IV* and *OOV* show results only for in-vocabulary and out-of-vocabulary words, respectively.

| Lexical Units | All | IV | OOV |
|---|---|---|---|
| Baseline | 0.2186 | 0.2864 | 0.0000 |
| +Morfessor | 0.2589 | 0.3135 | 0.0830 |
| +3gram-wi | 0.2586 | 0.3029 | 0.1160 |
| +5gram-wi | 0.2508 | 0.2988 | 0.0958 |
| +7gram-wi | 0.2517 | 0.2959 | 0.1094 |
| +3gram-cw | 0.2529 | 0.2951 | 0.1168 |
| +5gram-cw | 0.2466 | 0.2941 | 0.0932 |
| +7gram-cw | 0.2444 | 0.2906 | 0.0952 |

**Table 3**. ATWV results for the baseline word-based system combined with each sub-word based system individually.

### 4.2. Combined Results

We use a simple approach for system combination. The score for any keyword hypothesis is averaged over all systems that also make the same hypothesis. If only one system contains that hypothesis, then its score is used unaltered. In Table 3 we show the performance of every system combined with the baseline individually. Unexpectedly, even though it had the lowest OOV score, the Morfessor-based system provides the best combined performance. The improvement in ATWV over all keywords comes largely from the 9% relative improvement in IV keywords. In fact, all systems show an improvement in IV score when combined with the baseline system. This is surprising because the individual IV performance for some of the systems was significantly worse (up to 20% lower ATWV). Depending on the system the improvement in ATWV over all keywords is between 12% and 18% relative.

| Lexical Units | All | IV | OOV |
|---|---|---|---|
| Baseline+Morfessor | 0.2589 | 0.3135 | 0.0830 |
| +3gram-wi | 0.2734 | 0.3156 | 0.1377 |
| +5gram-wi | 0.2711 | 0.3156 | 0.1276 |
| +7gram-wi | 0.2745 | 0.3172 | 0.1368 |
| +3gram-cw | 0.2718 | 0.3126 | 0.1401 |
| +5gram-cw | 0.2699 | 0.3149 | 0.1249 |
| +7gram-cw | 0.2652 | 0.3099 | 0.1211 |

**Table 4**. ATWV results for the baseline word-based and Morfessor system combined with each sub-word based system individually.

For the final set of system combination results, Table 4 shows the effect of adding a third system to the two best performing systems—the baseline and the Morfessor-based sub-word system. Adding any of the ngram-based sub-word models produces a further improvement. Results are similar across all systems (20% to 25% relative ATWV improvement compared to the baseline).

## 5. ANALYSIS

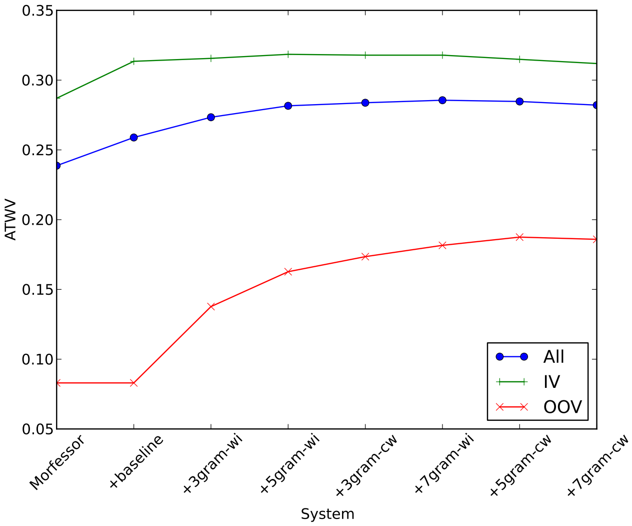

**Fig. 1**. ATWV performance change by iteratively adding each of the systems.

The combination of multiple systems significantly improves over baseline performance. Since each proposed sub-word unit requires a separate recognition system, the computational effort is linear in the number of systems. In Figure 1, we examine the benefit of continuing to combine more systems. The systems are ranked in terms of ATWV and at each step we add the next highest ranked system. We continue to obtain increasingly smaller improvements until we reach the sixth system. Going beyond six systems begins to slightly degrade performance.

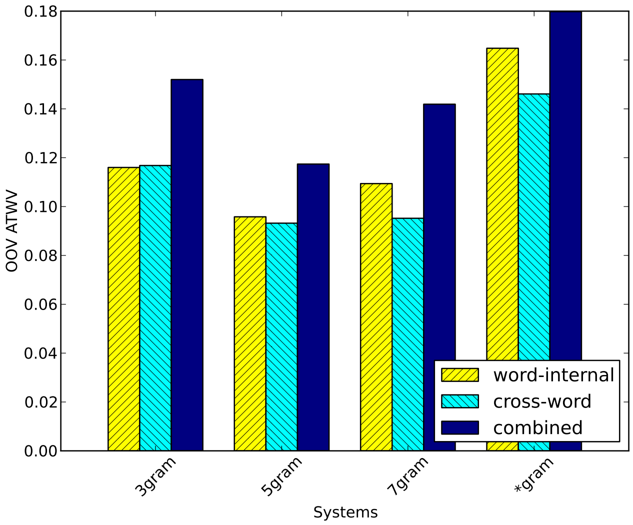

**Fig. 2**. Comparing the OOV ATWV performance of the word-internal units, cross-word units, and their combination.

The cross-word sub-word units provide similar performance to their word-internal counterparts. Figure 2 shows the ATWV on OOV keywords when combining the various ngram sub-word systems. Clearly the combinations are beneficial, so the cross-word models must add information not found in the systems without cross-word models. Though not in the figure, it is also interesting to note that by combining two or more sub-word systems, it is possible to outperform the baseline system on its own; the combination of *3gram-wi* and *3gram-cw* gives an 11% relative improvement over the baseline. Based on the results, we cannot say that cross-word sub-word units outperform other units, but they certainly provide additional information.

Wegmann et al. [24] performed a similar analysis of combining KWS systems, though the lexical units were identical for each system. They found that the increase in performance came mainly from the addition of a single true hypothesis for several keywords; we find a similar pattern in our results. All systems have a similar number of total correct detections. Comparing the best performing system and the baseline system, there is approximately a 6% absolute difference. The best performing system correctly detects 76 keywords with only a single entry in the development set that were missed by the baseline. The additional detection of these rare, missed keywords accounts for more than two-thirds of the difference in performance.

## 6. CONCLUSION

We have investigated various sub-word units for detecting OOV keywords. Our results show a relative gain in ATWV of 18% when combining the best two systems, and a gain of 25% when combining the best three systems. The gains are not due solely to the detection of OOV keywords, but also due to the increased detection of IV keywords. While most work in sub-word models focuses on OOV keywords, it is clear they can be useful for IV keywords too. Cross-word sub-word units were contrasted with word-internal sub-word units. While performance of the various sub-word units was similar, the cross-word sub-word units do combine well with the other units to produce additional gains.

We see two potential extensions to this work. The first is the development of more sophisticated, potentially keyword-specific, methods of combining the keyword hypotheses. In this work, we simply averaged over the various systems. Each of the systems likely perform better on certain keywords based on the structure of the lexical units; the system combination could consider this. The second extension would be to reduce the necessity of multiple systems. Instead, all of the possible sub-word units could be combined into a single, larger system.

## 7. ACKNOWLEDGEMENTS

This research was partially supported by OSEO, the French State agency for innovation, under the Quaero program and 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.

## 8. REFERENCES

[1] J. S. Bridle, “An efficient elastic-template method for detecting given words in running speech,” in *Proceedings of the British Acoustic Society Meeting*, 1973.

[2] J. R. Rohlicek, W. Russell, S. Roukos, and H. Gish, “Continuous hidden markov modeling for speaker-independent word spotting,” in *Proceedings of ICASSP*, 1989, pp. 627–630.

[3] D. A. James and S. J. Young, “A fast lattice-based approach to vocabulary independent wordspotting,” in *Proceedings of IEEE ICASSP*, 1994, pp. 377–380.

[4] A. G. Hauptmann, R. E. Jones, K. Seymore, S. T. Slattery, M. J. Wittbrock, and M. A. Siegler, “Experiments in information retrieva from spoken documents,” in *Proceedings of DARPA Workshop on Broadcast News Transcription and Understanding*, 1998, pp. 175–181.

[5] B. Kingsbury, J. Cui, X. Cui, M. J. F. Gales, K. Knill, J. Mamou, L. Mangu, D. Nolan, M. Picheny, B. Ramabhadran, R. Schluter, A. Sethy, and P. C. Woodland, “A high performance cantonese keyword search system,” in *Proceedings of ICASSP*, 2013, pp. 8277–8281.

[6] D. Karakos, R. Schwartz, S. Tsakalidis, L. Zhang, S. Ranjan, T. Ng, R Hsiao, G. Saikumar, I. Bulyko, L. Nyugen, 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 *Proceedings of IEEE ASRU*, 2013.

[7] P. Woodland, S. E. Johnson, P. Jourlin, and K. S. Jones, “Effects of out of vocabulary words in spoken document retrieval,” in *Proceedings of ACM SIGIR*, 2000, pp. 372–374.

[8] O. Siohan and M. Bacchiani, “Fast vocabulary-independent audio search using path-based graph indexing,” in *Proceedings of Interspeech*, 2005, pp. 53–56.

[9] U. V. Chaudhari and M. Picheny, “Improvements in phone based audio search via constrained match with high order confusion estimates,” in *Proceedings of IEEE ASRU*, 2007, pp. 665–670.

[10] M. Saraclar and R. Sproat, “Lattice-based search for spoken utterance retrieval,” in *Proceedings of HLT-NAACL*, 2004.

[11] G. Chen, O. Yilmaz, J. Trmal, D. Povey, and S. Khudanpur, “Using proxies for OOV keywords in the keyword search task,” in *Proceedings of IEEE ASRU*, 2013, pp. 416–421.

[12] V. T. Turunen and M. Kurimo, “Speech retrieval from unsegmented finnish audio using statistical morpheme-like units for segmentation, recognition, and retrieval,” *ACM Transactions on Speech and Language Processing*, vol. 8, no. 1, pp. 1–25, 2011.

[13] I. Szoke, L. Burget, J. Cernocky, and M. Faspo, “Sub-word modeling of out of vocabulary words in spoken term detection,” in *Proceedings of IEEE Workshop on Spoken Language Technology*, 2008, pp. 273–276.

[14] I. Bulyko, J. Herrero, C. Mihelich, and O. Kimball, “Subword speech recognition for detection of unseen words,” in *Proceedings of Interspeech*, 2012.

[15] S. Virpioja, P. Smit, S.-A. Gronroos, and M. Kurimo, “Morfessor 2.0: Python implementation and extensions

for morfessor baseline," Tech. Rep., Aalto University, 2013.

[16] K.-S. Cheng, G. H. Young, and K.-F. Wong, "Study on word-based and integral-bit chinese text compression algorithms," *Journal of the American Society for Information Science*, vol. 50, no. 3, pp. 218–228, 1999.

[17] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddington, "Results of the 2006 spoken term detection evaluation," in *Proceedings of ACM SIGIR*, 2007, pp. 51–55.

[18] 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 *Proceedings of IEEE ASRU*, 2011.

[19] D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Glembek, N. Goel, M. Karafiat, A. Rastrow, R. Rose, P. Schwarz, and S. Thomas, "The subspace gaussian mixture model - a structured model for speech recognition," *Computer Speech and Language*, vol. 25, pp. 404–439, 2011.

[20] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space disciminative training," in *Proceedings of IEEE ICASSP*, 2008, pp. 4057–4060.

[21] R. Kneser and H. Ney, "Improved backing-off for n-gram language modeling," in *Proceedings of IEEE ICASSP*, 1995, pp. 181–184.

[22] D. Can and M. Saraclar, "Lattice indexing for spoken term detection," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2338–2347, 2011.

[23] D. R. H. Miller, M. Kleber, C. Kao, O. Kimball, T. Colhurst, S. Lowe, R. M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Proceedings of Interspeech*, 2007, pp. 314–317.

[24] S. Wegmann, A. Faria, A. Janin, K. Riedhammer, and N. Morgan, "The tao of ATWV: Probing the mysteries of keyword search performance," in *Proceedings of IEEE ASRU*, 2013, pp. 192–197.