Comparing Decoding Strategies for Subword-based Keyword Spotting in Low-Resourced Languages

William Hartmann[1], *Viet-Bac Le*[2], *Abdel Messaoudi*[2], *Lori Lamel*[1], *Jean-Luc Gauvain*[1]

[1]CNRS/LIMSI, Spoken Language Processing Group, 91403 Orsay cedex, France
[2]Vocapia Research, 28 rue Jean Rostand, 91400 Orsay, France
{hartmann, lamel, gauvain}@limsi.fr, {levb, abdel}@vocapia.com

Abstract

For languages with limited training resources, out-of-vocabulary (OOV) words are a significant problem, both for transcription and keyword spotting. This paper investigates the use of subword lexical units for keyword spotting. Three strategies for using the sub-word units are explored: 1) converting word-based lattices to subword lattices after decoding, 2) performing a separate decoding for each subword type, and 3) a single decoding using all possible subword units. In these experiments, the best performance is achieved by carrying out a separate decoding for each subword type. Further gains are attained through system combination. We also find that ignoring word boundaries improves the detection of OOV keywords without significantly impacting in-vocabulary keyword detection. Results are presented on four languages from the IARPA Babel Program (Haitian Creole, Assamese, Bengali, and Zulu).
Index Terms: keyword search, spoken term detection, OOV, sub-word lexical units, low resource LVCSR

1. Introduction

Keyword spotting (KWS)—or spoken term detection (STD)—searches audio for a specific keyword—keywords can also be sequences of words. The task differs from the traditional speech-to-text (STT) task in that only a subset of words are important. For STT, every spoken word is equally important for the word error rate (WER) metric.

Out-of-vocabulary (OOV) keywords are a significant problem for KWS. While the OOV rate may be relatively low for a language, OOV words can represent a disproportionate number of keywords. Many methods have been previously proposed for detecting OOV keywords. A typical approach is to convert word lattices to phone lattices [1, 2]. Keywords are then detected by searching for a matching phone string. Converting the lattice to subwords is another alternative [3, 4]. Instead of converting the word lattice, additional hypotheses can be considered by using a phonemic confusion matrix [5]. Chen et al. take a similar approach, but from the keyword perspective [6]. When a keyword is not in the vocabulary, the search is expanded to include phonetically similar words in the original vocabulary.

These approaches all attempt to recover OOV words after decoding has already been performed. The OOV terms can also be anticipated by decoding with lexical units that are more likely to detect OOV terms. Previous work has used both phones as lexical units [7] and character ngrams [8]. Different types of subword units can also be combined and detected in a single decoding step [9].

In this work, we focus on the subword-based approach to KWS. Our goal is to examine three methods for using the sub-word units in the detection process: 1) converting word lattices to subword lattices, 2) using a separate decoding pass for each type of subword unit, and 3) combining all subword units together and performing a single decoding. Since the final method produces a lattice containing all possible subword units, we also simulate a subword unit-dependent decoding by only allowing a subset of subword units when searching for keywords.

Section 2 introduces the IARPA Babel data. Section 3 introduces the keyword spotting system. A detailed description of the decoding strategies is presented in Section 4. Section 5 presents the results and comparisons, and the concluding remarks are in Section 6.

	IV Keywords		OOV Keywords	
Language	in Dev	not in Dev	in Dev	not in Dev
Creole	2644	1055	382	615
Assamese	2402	1439	682	530
Bengali	2442	1300	761	550
Zulu	1740	1196	1477	737

Table 1: Distribution of keywords for each of the languages. Note that a large number of keywords are not found in the development data and have no effect on the final results.

2. IARPA Babel Data

2.1. Data Description

All experiments in this work use data from the IARPA-funded Babel program. The languages considered are Haitian Creole (iarpa-babel201b-v0.2b), Assamese (iarpa_babel_op1_103), Bengali (iarpa_babel_op1_102), and Zulu (iarpa-babel206b-v0.1d). For each language, only the 10-hour subset of transcribed training data (limited language pack condition) is used for training acoustic and language models. Results are reported on a separate 10-hour development set.

For the keyword spotting experiments, we use the official evaluation keywords plus an additional set of development keywords. Table 1 provides an analysis of the keywords for each language in terms of whether they occur in the training (IV) or development data. A keyword can consist of a sequence of words and is case-insensitive. In the case of multi-word keywords, if any of the individual words are OOV, then the entire keyword is considered OOV. If each of the individual words are seen in training, the keyword is considered IV, even if that exact sequence of words were not seen.

Zulu is the most challenging of the four languages. In addition to the complex morphology, it is tonal and has clicks as part of its phonology. Both Assamese and Bengali share a non-

Roman script—with minor variations. Haitian Creole is similar to French and has a limited morphology.

2.2. Performance Metric

Maximum term-weighted value (MTWV) and actual term-weighted value (ATWV) are defined as the measures of interest for IARPA Babel program. ATWV was also used in the NIST 2006 Spoken Term Detection evaluation [10]. The keyword specific ATWV for keyword k at a specific threshold t can be computed by

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

where P_{FR} and P_{FA} refer to the probability of a false reject (miss) and false accept, respectively. The constant β—set to a value of 999.9—mediates the trade off between false accepts and false rejects. MTWV represents the maximum score that could be obtained by using the optimal value for t over all keywords. We report results on MTWV in this work, but the differences between ATWV and MTWV are small. The range for MTWV (and ATWV) is between $-\infty$ and 1; incorrect hypotheses are worse than no hypotheses.

In this performance metric, every keyword is equally weighted regardless of its frequency. Missing a single occurrence of a rare word can affect the final score as much as missing a more common word dozens of times. This explains why so much effort is devoted to the detection of OOV keywords. Wegmann et al. have a more detailed discussion of ATWV and MTWV in relation to the IARPA Babel program [11].

3. Keyword Spotting System

3.1. Acoustic Model Training

The LIMSI STK toolkit [12] was used to train and decode all systems. All voice activity detection (VAD) was performed by the BBN VAD system [13]. The BUT stacked bottleneck features [14] are also used. More details about the acoustic models can be seen in [15]. Language models are built using only the 10 hours of transcribed training data. A bigram model is used to generate the initial lattice, but a 3-gram language model is used when converting to the consensus network.

In order to remove the necessity of a pronunciation lexicon and to easily generate pronunciations for subword units, we use graphemes as our acoustic units. In addition, we use position-independent acoustic units so that the same acoustic model can be used for all subword types. The number of graphemes for each language are: Haitian Creole, 31; Assamese, 49; Bengali, 48; and Zulu, 26. Each system also uses three additional acoustic units for non-speech (silence, breath, and filler words).

WER is quite high for the Babel data [16], even using state-of-the-art ASR systems and techniques. This is not unexpected as the data consists of conversational telephone speech with limited training data. When using the acoustic models described in this paper for transcription, the WER ranges from 51.5% for Creole to 65.2% for Zulu [15].

3.2. Keyword Spotting

Using a bigram language model, a single pass decoding generates a lattice. The lattice is rescored with a trigram language model and converted to a consensus network (CN) [17] prior to the keyword search. Our preliminary experiments found better OOV performance using the CN compared to the lattice; the CN contains sequences of words that do not exist in the original lattice. Since many of the keywords are actually sequences of words, we search the CN for any matching sequence of words. Up to 50ms of silence are permitted between any two words. In order to detect more OOV keywords, we also ignore word boundaries when searching for matches. As per the guidelines of the task, keyword matches are case-insensitive. All experiments follow the no test audio re-use (NTAR) condition; all audio is processed before the keywords are known [18].

The CN contains a posterior value for each hypothesis. For multi-word keywords, we use the geometric mean of the individual posteriors. Before evaluating the results, we apply the BBN score normalization procedure [19]. Score normalization significantly improves results and brings the ATWV results near the MTWV results.

3.3. Subword Units

We use two main types of subword units. Note that we also explored using single character graphemes (a common approach in the literature [1]), but preliminary results on Bengali were poor, so we did not explore it further. The first type of units are created by Morfessor [20], a tool for morphological decomposition. Given a set of words and their frequency, Morfessor learns a generative model that uniquely decomposes any word into a sequence of morphological units.

The second type of subword unit is based on character ngrams. To generate these units, two properties are first defined—the maximum length of any subword unit and whether cross-word subword units are considered. Given those two properties, a set of all possible subword units are constructed and used to build a uniform language model. The training corpus is segmented using the uniform language model. As all words are equally probable, this is equivalent to minimizing the total number of units in the segmentation. A new trigram language model is built from the segmented training corpus, and the corpus is resegmented. This process is repeated until convergence. Unlike the previous subword type the segmentation for any word is not necessarily unique; the segmentation depends on the surrounding context.

This work considers character ngrams of length 3, 5, and 7. The word-internal subword units are referred to as *3gram-wi*, *5gram-wi*, and *7gram-wi* throughout the remainder of the paper. Cross-word subword units are referred to as *3gram-cw*, *5gram-cw*, and *7gram-cw*. These subword units were also used in a previous study that demonstrated the efficacy of cross-word subword units on Turkish data [21].

4. OOV Keyword Detection Approaches

4.1. Lattice Mapping

One approach to OOV keyword detection is to perform recognition with the standard word units and then map the lattice to subword units. To test this approach, we use both types of subword units described in Section 3.3. For the subword units based on character ngrams, we only consider the word-internal versions. As mentioned previously, the segmentation may not be unique for each word, so we only use the most likely segmentation. After the words in the lattice have been converted to the subword units, the lattice is converted to a CN. One of the major benefits of this approach is that many types of units can be examined with a single decoding of the data. However, there is a computational drawback. If the subword units are short, the final CN will contain large numbers of short units. Given a large number

Language	Word Boundary	All	IV	OOV
Creole	keep	0.4310	0.4933	0.0000
Creole	remove	0.4554	0.4996	0.1518
Assamese	keep	0.2667	0.3425	0.0000
Assamese	remove	0.2884	0.3496	0.0737
Bengali	keep	0.2440	0.3200	0.0000
Bengali	remove	0.2741	0.3288	0.1014
Zulu	keep	0.1906	0.3523	0.0000
Zulu	remove	0.2280	0.3545	0.0806

Table 2: MTWV results comparing the effects of ignoring word boundaries when detecting keywords. All results use words as lexical units.

of keywords, particularly if they are long, it can take significant time to search the CN for the keywords.

4.2. Subword Unit-Dependent Decoding

An alternative approach is perform a separate decoding for each type of subword unit. After the training corpus has been segmented, a language model is built using the subword units. Assuming the same training process, a language model based on subword units obviously contains less context—negatively impacting WER—but it allows for the recognition of words not seen in training. There are approaches to compensate for this effect [22], but they are not explored in this work. Based on this language model, a lattice is decoded and converted to a CN.

The drawback to this approach is the computational cost. The additional cost is linear in the number of subword unit types. However, since the acoustic units are position-independent, increasing the types of subword units does not affect training time.

4.3. Joint Subword Unit Decoding

We can reduce the computational cost of using subword units by combining all subword unit types into a single system. Given multiple segmentations of the training transcript—one for each subword unit type—a single language model is built over all segmentations. We test two variants of this approach. The first only includes subword units, while the second also allows the inclusion of the original word units.

We also attempt to simulate subword unit-dependent decoding with the joint subword systems. Given a set of subword units, all other units in the CN are removed; the additional probability mass is distributed among the remaining hypotheses in proportion to their original confidence score. While this gives the ability to only consider certain types of units during search, it is not equivalent to unit-dependent decoding.

5. Results

5.1. Word Boundaries

Before comparing the performance of the subword units and decoding strategies, we show baseline performance and investigate the effects of word boundaries. Table 2 lists two results per language. All results use words as lexical units. The difference is whether word boundaries are considered. When word boundaries are ignored, a significant portion of the OOV keywords are detected—with little effect on IV keywords. The pattern for subword units is similar, though the gain for OOV keywords is smaller. Based on these results, the remaining experiments will search for keywords while ignoring word boundaries.

Language	Lex. Unit	Conversion	Dependent	Simulated
Creole	morfessor	0.2076	0.2994	0.2257
Creole	3gram-wi	0.2139	0.3349	0.2813
Creole	3gram-cw	_____	0.3439	0.2763
Creole	5gram-wi	0.2181	0.3361	0.2815
Creole	5gram-cw	_____	0.3484	0.3196
Creole	7gram-wi	0.2223	0.3630	0.3030
Creole	7gram-cw	_____	0.3890	0.3229
Assamese	morfessor	0.0765	0.1280	0.0978
Assamese	3gram-wi	0.0899	0.1302	0.0941
Assamese	3gram-cw	_____	0.1430	0.0996
Assamese	5gram-wi	0.0842	0.1648	0.1139
Assamese	5gram-cw	_____	0.1348	0.1143
Assamese	7gram-wi	0.0898	0.1669	0.1169
Assamese	7gram-cw	_____	0.1562	0.1123
Bengali	morfessor	0.0918	0.1622	0.1034
Bengali	3gram-wi	0.1184	0.1774	0.1151
Bengali	3gram-cw	_____	0.1433	0.1159
Bengali	5gram-wi	0.1147	0.1819	0.1328
Bengali	5gram-cw	_____	0.1664	0.1329
Bengali	7gram-wi	0.1145	0.1792	0.1328
Bengali	7gram-cw	_____	0.1646	0.1326
Zulu	morfessor	0.0845	0.2539	0.2114
Zulu	3gram-wi	0.1026	0.2904	0.1772
Zulu	3gram-cw	_____	0.2736	0.1926
Zulu	5gram-wi	0.0666	0.2817	0.2581
Zulu	5gram-cw	_____	0.2846	0.2669
Zulu	7gram-wi	0.0661	0.2809	0.2652
Zulu	7gram-cw	_____	0.2931	0.2658

Table 3: MTWV results on OOV keywords. Conversion refers to converting words in the lattice to subword units prior to searching. Dependent refers to using a separate decoding per subword unit type. Simulated refers to performing decoding with all subword units, but limiting the search to a specific type (see Section 4.3). Note that we do not convert words to cross-word subword units, so no results exist for those cases.

5.2. Lattice Conversion vs. Unit-Dependent Decoding

Table 3 compares the performance of several types of subword units when converting the lattice or performing unit-dependent decoding. Due to space constraints, only performance on OOV keywords are shown. No subword unit gives better performance than baseline word decoding for IV keywords. As expected, performing a separate decoding for each type of subword unit gives significantly better performance—ranging from 50% to over 100% relative improvement. It is interesting to note the subword-based units do not perform as well for Assamese and Bengali, possibly due to their larger character and grapheme set.

In all cases, the worst performing subword type is the Morfessor-based subword type, but this is misleading. The Morfessor-based units consistently give the best IV keyword performance—though still not as good as the baseline word system—and can give the best overall performance depending on the ratio of IV to OOV keywords. When comparing the character ngram-based subword unit results across languages, there does not appear to be a pattern. The worst system for Creole (*3gram-wi*) performs well for both Bengali and Zulu.

In a previous study on Turkish [21], we noted that cross-word subword units did not outperform word-internal units, but

Language	Lex. Unit	All	IV	OOV
Creole	joint	0.4625	0.4788	0.3556
Creole	joint+word	0.4561	0.4728	0.3440
Assamese	joint	0.2868	0.3297	0.1424
Assamese	joint+word	0.2862	0.3310	0.1415
Bengali	joint	0.2668	0.3000	0.1652
Bengali	joint+word	0.2668	0.2983	0.1657
Zulu	joint	0.3041	0.3256	0.2844
Zulu	joint+word	0.3049	0.3350	0.2795

Table 4: MTWV results for the joint decoding systems. Joint refers to a system that decodes using all possible subword units. Joint+word also includes the original words.

Language	Decode Type	All	IV	OOV
Creole	Conversion	0.4173	0.4507	0.2498
Creole	Dependent	0.4664	0.4734	0.4319
Creole	Simulated	0.4368	0.4590	0.3135
Assamese	Conversion	0.2670	0.3154	0.1003
Assamese	Dependent	0.2907	0.3228	0.1859
Assamese	Simulated	0.2762	0.3212	0.1276
Bengali	Conversion	0.2601	0.3012	0.1288
Bengali	Dependent	0.2858	0.3015	0.2370
Bengali	Simulated	0.2613	0.2969	0.1489
Zulu	Conversion	0.1862	0.2802	0.0878
Zulu	Dependent	0.3132	0.3110	0.3221
Zulu	Simulated	0.2420	0.2598	0.2301

Table 5: MTWV results for the combined results from Table 3.

did combine well. For these results, the cross-word units outperform their word-internal counterparts in several instances; *7gram-cw* gives the best result for both Creole and Zulu. It may be that cross-word units work best with longer subword units.

5.3. Joint Subword Unit Decoding

Table 4 shows the performance of using a single decoding for all possible subword units. Results are presented both with and without the inclusion the original words. The inclusion of the original words has little effect on the final performance—the combination of all the subword units actually cover a significant portion of the original words anyway. As a stand alone system, the joint decoding is several points worse than the best subword unit-dependent system on OOV, and several points worse than the word-based system on IV. However, depending on the balance of IV to OOV, it can offer the best combined performance. Since the Zulu keyword list contains so many OOV keywords, the joint Zulu result gives a 34% relative increase in MTWV.

In Section 4.3 we also discussed simulating the unit-dependent systems with a single joint system. Table 3 contains the results. In all cases, this approach of simulating the unit-dependent decoding is significantly worse than actually performing the unit-dependent decoding. However, it is significantly better than converting the original word lattice to the equivalent subword unit. This approach could offer a way to quickly generate large numbers of additional hypotheses depending on the subword units considered.

5.4. Combination Results

We use a simple procedure for combining results—overlapping hypotheses are averaged. More sophisticated combination methods could produce better results, but this provides a guide to the complementarity of the results. Table 5 contains the results for combining the results for each decoding strategy discussed in Section 4.

The lattice conversion results are at a slight disadvantage because the combination does not include the cross-word subword units, but it is clearly the worst performing approach. The simulated results offer better performance than the lattice conversion, but fail to obtain the performance of the joint system used to generate them. The best performance is from combining the subword unit-dependent results.

6. Conclusion

A major difficulty in the KWS task is the detection of OOV keywords. Subword-based lexical units are one potential solution. We have investigated three approaches to incorporating subword units into the decoding process. The simplest, converting a word lattice to subword units, also performs the worst. While it can detect some OOV keywords, the sequences of subword units that can form OOV words do not appear often.

The second approach, subword unit-dependent decoding, provides the best OOV performance, especially when combining multiple results. However, it does come at a higher computational cost since multiple decoding passes are required. Also, it is not clear if determining which subword unit types will perform best prior to decoding is possible.

The final approach, joint subword decoding, has worse IV performance than the baseline and worse OOV performance than the best subword unit-dependent result. However, depending on the percentage of OOV keywords, it can produce the best overall result. If a single system is required, it is a good candidate. We also explored the idea of simulating the unit-dependent systems using the joint results. The simulated performance is not as good, but provides a method for creating a large set of results that can later be combined.

We find the different subword unit results contain complementary information. Even using a simple combination approach, large gains in MTWV are seen. We plan to explore more sophisticated combination methods to improve results even further. The Bengali and Assamese gains were not as large as with Creole and Zulu. In future work we will investigate whether this is due to a specific characteristic of these languages, and whether an improvement to the subword unit-based methods can compensate for it.

7. Acknowledgements

We would like to thank other partners of the BABELON team on the IARPA-Babel project for exchanging the resources (especially BUT for the bottle-neck features, BBN for VAD, autotranscription, and score normalization tool). We would also like to acknowledge the help and contribution of other colleagues in LIMSI and Vocapia.

This research was supported 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] O. Siohan and M. Bacchiani, "Fast vocabulary-independent audio search using path-based graph indexing," in *Proceedings of Interspeech*, 2005, pp. 53–56.

[2] D. Karakos, I. Bulyko, R. Schwartz, S. Tsakalidis, L. Nyugen, and J. Makhoul, "Normalization of phonetic keyword search scores," in *Proceedings of IEEE ICASSP*, 2014.

[3] 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.

[4] Y. He, B. Hutchinson, P. Baumann, M. Ostendorf, E. Fosler-Lussier, and J. Pierrehumbert, "Subword-based modeling for handling OOV words in keyword spotting," in *Proceedings of IEEE ICASSP*, 2014.

[5] P. Karanasou, L. Burget, D. Vergyri, M. Akbacak, and M. Arindam, "Discriminatively trained phoneme confusion model for keyword spotting," in *Proceedings of Interspeech*, 2012, pp. 2434–2437.

[6] 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.

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

[8] 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.

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

[10] 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.

[11] 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.

[12] J. L. Gauvain, L. Lamel, and G. Adda, "The LIMSI broadcast news transcription system," *Speech Communication*, vol. 37, no. 1-2, pp. 89–108, 2002.

[13] T. Ng, B. Zhang, L. Nyugen, S. Matsoukas, X. Zhao, N. Mesgarani, K. Vesely, and P. Matejka, "Developing a speech activity detection system for the DARPA RATS program," in *Proceedings of Interspeech*, 2012, pp. 1969–1972.

[14] F. Grézl and M. Karafiát, "Semi-supervised bootstrapping approach for neural network feature extractor training," in *Proceedings of IEEE ASRU*, 2013, pp. 470–475.

[15] V.-B. Le, L. Lamel, A. Messaoudi, W. Hartmann, J. L. Gauvain, C. Woehrling, J. Despres, and A. Roy, "Developing STT and KWS systems using limited language resources," in *Proceedings of Interspeech*, 2014.

[16] S. Tsakalidis, R. Hsiao, D. Karakos, T. Ng, S. Ranjan, G. Saikumar, L. Zhang, L. Nyugen, R. Schwartz, and J. Makhoul, "The 2013 BBN vietnamese telephone speech keyword spotting system," in *Proceedings of IEEE ICASSP*, 2014.

[17] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: Word error minimization and other applications of confusion networks," *Computer Speech and Language*, vol. 14, no. 4, pp. 373–400, 2000.

[18] NIST, "The OpenKWS14 Evaluation Plan, v11," http://www.nist.gov/itl/iad/mig/upload/KWS14-evalplan-v11.pdf, December 2013.

[19] 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.

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

[21] W. Hartmann, L. Lamel, and J. L. Gauvain, "Cross-word subword units for low-resource keyword spotting," in *SLTU*, 2014, pp. 112–117.

[22] M. Kurimo, A. Puurula, E. Arisoy, V. Siivola, T. Hirsimäki, J. Pylkkönen, T. Alumäe, and M. Saraclar, "Unlimited vocabulary speech recognition for agglutinative languages," in *Proceedings of HLT-NAACL*, 2006, pp. 487–494.