CHARACTER-BASED UNITS FOR UNLIMITED VOCABULARY CONTINUOUS SPEECH RECOGNITION

Peter Smit, Siva Reddy Gangireddy, Seppo Enarvi, Sami Virpioja, Mikko Kurimo

Department of Signal Processing and Acoustics, Aalto University, Finland

ABSTRACT

We study character-based language models in the state-of-the-art speech recognition framework. This approach has advantages over both word-based systems and so-called end-to-end ASR systems that do not have separate acoustic and language models. We describe the necessary modifications needed to build an effective character-based ASR system using the Kaldi toolkit and evaluate the models based on words, statistical morphs, and characters for both Finnish and Arabic. The morph-based models yield the best recognition results for both well-resourced and lower-resourced tasks, but the character-based models are close to their performance in the lower-resource tasks, outperforming the word-based models. Character-based models are especially good at predicting novel word forms that were not seen in the training data. Using character-based neural network language models is both computationally efficient and provides a larger gain compared to the morph and word-based systems.

Index Terms— speech recognition, subword-based language modeling, neural network language models, low resource, unlimited vocabulary

1. INTRODUCTION

The lexicon and language models for large-vocabulary continuous speech recognition (LVCSR) systems for western languages are still typically built using words as the basic units. However, a lot of research has been published on alternative subword units such as morphemes or data-driven segments [1]. For agglutinative languages such as Finnish these subword models provide the state-of-the-art in LVCSR [2, 3].

Recently, end-to-end speech recognition systems have been created that use the smallest unit of written language; single graphemes [4, 5]. These systems have been built in such a fashion that the borders between acoustic and language models disappear, and that the whole system can be trained using a single dataset. A lot of techniques introduced in the end-to-end speech recognition systems have created insights for making conventional speech recognition systems even better. For example, a state-of-the-art toolkit such as Kaldi [6] uses frame subsampling and sequence-based training which are directly inspired by end-to-end systems.

To our knowledge, there has been no work published on using only characters as units in a conventional speech recognition system for languages with phonemic orthography. In this work, we implement a character-based speech recognizer in a conventional state-of-the-art speech recognition framework. We applied the proposed character-based approach in the Arabic MGB-3 challenge [7], where our system was the winning entry with the lowest word error rate. In this paper, we explore the character-based models also for Finnish, as well as for more low-resourced scenarios, where the approach has the greatest advantages. We also explore in more detail the performance and the properties of character-based systems.

Section 2 gives a background on earlier subword-based systems. In Section 3 the prerequisites and implementation details of our system are described. Sections 4, 5, and 6 analyze our system in general ASR performance, the ability to predict unseen words, and a comparison of word, subword and character models in regard of system requirements for each part of the speech recognition pipeline.

2. SUBWORD LANGUAGE MODELING

The modeling of the vocabulary in LVCSR systems has been an active research topic for a long time [8]. Whereas for languages like English it is possible to get a reasonable coverage of a language by taking e.g. the most frequent 100,000 words, for other languages millions of words would be needed to cover a similar part. The first effective LVCSR systems often had vocabularies ranging from 1,000 to 20,000 word types [9], because obtaining improvements via larger vocabularies were beyond the computers' memory and decoding capacity at the time. However, in highly inflectional or agglutinative languages the amount of commonly used word forms can be much higher, which led to the search of alternative models for improving the coverage of the language. Multiple techniques were developed such as class-based language models [10] and the use of subword units [1] or mixture of word and subword units [11].

Subword models are created by splitting the words in a

This work was financially supported by the Academy of Finland under the grant number 251170 and TELLme-project in Tekes Challenge Finland programme. PS would like to thank Lingsoft, Inc. and the Foundation for Aalto University Science and Technology. Computational resources were provided by the Aalto Science-IT project.

systematic manner. This can be either linguistically inspired, such as stemming or morphological segmentation, or data-driven. Using linguistics requires expert knowledge to create the segmentation; the data-driven way often requires parameter tuning to control the size of the subword vocabulary.

The subword models have multiple pros and cons. Whereas in the past a smaller vocabulary size was important for the whole LVCSR system to run, it is now beneficial for the state-of-the-art Neural Network Language Models (NNLMs). Since the input and output layer dimensions depend on the vocabulary size, very large vocabularies are unpractical. Although the layers can be reduced by shortlists and class-based models [12, 13], or computations approximated by methods like hierarchical softmax [14], the subword models provide a natural and effective way to lower the dimensionality.

Besides having a smaller vocabulary size, the subword models also increase the coverage of the language. A properly chosen set of subword units can model any words in a language by concatenating the units. In practice, this means that there will be no out-of-vocabulary (OOV) words and that language models can effectively adapt to texts with previously unseen words. This can be important, because even if all the words in the training corpus could be taken in the vocabulary, many of them appear only once or twice in the data. This sparsity causes poor probability estimates, and especially the n-gram models will often have to back-off to unigram probabilities. The sparsity and the vocabulary coverage problems only get worse when the language and in-domain resources are more limited. A subword vocabulary is better covered in the data and has much more training samples per vocabulary item.

A drawback of subword models is the need for a long sequence of units to cover word and sentence context. This is particularly problematic for models, such as n-gram or feed-forward NNLMs, that predict words given a fixed window of units for history. For n-gram models this drawback can be alleviated by using techniques such as variable-order models [15, 16, 17]. For the state-of-the-art Recurrent NNLMs, modeling long sequences is even less problematic, because the model takes the complete context into account using the recurrent connections.

There has been also a lot of research on the use of 'hybrid models', which contain a mixture of both subword and word units [11, 18, 19]. These models can model the selected long words accurately and still cover the left over OOVs, but are structurally more complex for the LVCSR system.

2.1. Subwords to the extreme: Characters

Having a long experience on subword modeling challenged us to explore the extreme case of building a state-of-the-art LVCSR system using only single characters as language modeling units. There would be no need to train and tune a subword segmentation algorithm or use expert knowledge to generate a morphological analyzer. The resulting model would have by definition a full coverage of the vocabulary, as long as words are written in the same alphabet. For NNLMs, the dimensions of character input and output layers would be very small, giving room for experimenting with specific network architectures without the burden of high-dimensional input and output. Furthermore, it may become possible to train robust and successful models using even smaller amounts of data than for longer subword units, which is important for low-resource languages.

There have been interesting developments in the creation of 'end-to-end' speech recognition systems [5, 20] that create joint audio and language models, often working with the character as basic unit. In contrast to these systems, we explore the use of characters in a 'conventional' state-of-the-art LVCSR system. Exploring how character units work in conventional speech recognition systems will also give insight into the use of character units in the 'end-to-end' systems. The conventional systems have the advantage that language models can be separately trained and adapted to different domains without having transcribed acoustic data. This is particularly important when creating speech recognizers for domains or languages where little transcribed speech is available.

Character language models have been used as well in other fields of language technology, such as in machine translation [21, 22].

3. CHARACTER-MODEL IMPLEMENTATION

When modifying the LVCSR system to utilize character-based units, a number of changes in the implementation are needed to make the models and the recognition process efficient and allow the system to reconstruct words from the recognized character sequence. Most of these changes are analogous to the changes that are needed for implementing longer subwords, so we have built the system by modifying the techniques described in [3].

For acoustic model training no changes are needed. They can be trained in a regular way on a sequence of phonemes provided by the word-based transcripts. The units utilized in the decoding are not influenced by training process, as the training lexicon can be completely separate from the recognition lexicon. An important part of acoustic modeling is that the character-based models require a grapheme-based phoneset. This is usually not a problem for highly phonemic scripts, because the acoustic models with a grapheme-based lexicon can easily learn the small phonemic changes and variations using the immediate grapheme context. However, for a language such as English that is far from phonemic, this does pose an obstacle that requires further research.

For language model training the original training corpus is split into characters instead of words. In order to be able to reconstruct the word boundaries from the character sequences later, different marking schemes can be used for the characters. The most straightforward solution is to mark the word bound-

aries by adding a separate word boundary token. However, to avoid adding extra tokens, it is also possible to mark the grapheme at the begin or end of the word, or both. After splitting the words into characters and adding the word markers, the language models can be trained in the usual way. In this work we try all these four different marking styles, as previous research has shown that the selection of the best one is dependent on the data.

For n-gram models, special techniques are needed to create a character model that would be as accurate as a word or morph model trained on the same corpus. Often the order of a word n-gram model is restricted to a relatively low number, e.g. four or five. However, the shorter units we use, the higher order of n-gram model is required. For example, to be equivalent with a 4-gram word model in Finnish, a character model would need to contain contexts of order 20 or higher. To effectively train high-order n-gram models we use the VariKN toolkit [17], which can grow and prune variable-order Kneser-Ney smoothed n-gram models. This tool has been effective for various subword models in the past [16], and we show that it can successfully train high-order character n-gram models as well. The training of the character language model is computationally heavier than a word or morph model, which can be seen from the analyses given in Section 6.

Recurrent Neural Network Language Models (RNNLMs) have clearly dominated in performance over n-gram -based language models in the past years [23, 24]. As these models are specialized in capturing longer contexts, there is no specific modifications needed to work with character-based models, except to increase the maximum length of history that is considered during training. A significant advantage in the character models is the low dimensionality of the input and output layer. As the vocabulary of a character model is very small, a normal softmax output normalization can be used.

To combine the acoustic model, the language model and the grapheme lexicon containing only single characters into an efficient decoding graph, we use the modifications described in [3] to restrict the lexicon to give only legal sequences of tokens. For example, when word boundary markers are used, they have to appear at both the beginning and end of a sentences, and when word-continuation characters (so-called left-marked, +m) are used, the first character in a sentence has to be restricted to an unmarked character.

After decoding, there are two ways to reconstruct the word sequence from a character sequence. The first is to take the generated hypothesis string and apply string operations to reverse the marking scheme that was used in the language model training. The second is to transform the character-based lattice into a word lattice. This is especially useful when word-level post-processing, such as the system combination with a word model, is performed afterwards. For creating the word lattice, we use the following procedure: First, we identify all nodes in a lattice that indicate word boundaries. After that, for each word boundary node, we list all paths from the node until the next word-boundary node is found. These paths describe all words that can be generated by the lattice. For the transformation, a finite-state transducer that maps each sequence of character to its corresponding word is created and applied to the lattice.

4. SPEECH RECOGNITION EXPERIMENTS

To evaluate the speech recognition performance of character-based models we run a set of experiments for two languages: Finnish and Arabic. Finnish is a morphologically rich and agglutinative language that requires a very large vocabulary due to inflections, derivations, and compounding. In LVCSR there is a long history of using statistically created subword models to reduce the vocabulary size [1]. Arabic is also morphologically rich, but not an agglutinative language. In the past morpheme-based models have been used successfully for Arabic [25, 26, 27]. Besides comparing character and word models, we also create optimized subword models using the Morfessor toolkit [28, 29] and report those results as well. Furthermore, to simulate an under-resourced situation we also perform all experiments with models trained on only 10% of the available language modeling data.

4.1. Datasets

For Finnish we used three different corpora for acoustic model training. The Speecon corpus [30] consists of recorded speech under multiple conditions with multiple microphones, of which we used the lapel microphone. The Speechdat corpus contains read speech over a low-quality phone transmission [31]. Lastly the Parliament corpus contains speech by members of parliament during its sessions [32]. In total more than 1500 hours of data was used for acoustic model training. As test data we used a set of broadcast news from the Finnish national broadcaster (Yle). This test dataset has 5 hours of speech and 35k words. The same set of training corpora and test data was used in previous work [32].

For Arabic acoustic models we used the training corpus of the MGB-2 challenge [33], which consists of 1,200 hours of broadcast data from multiple genres and even dialects. As test data we used the MGB-2 dev-set, which has 8 hours of data and 57k words.

For language modeling in Finnish we used a corpus of newspaper texts from the Finnish Text Collection [34]. This corpus contains 160 million tokens, with 4.3 million unique words. Similarly, for Arabic we used a corpus of 130 million tokens crawled from the Al Jazeera website, which contains 1.4 million unique words. After reducing 90% of both corpora, the Finnish corpus had 1.3 million word types left and the Arabic corpus 380k word types.

4.2. Setup

The acoustic models were trained using the Kaldi toolkit [6] and have a bidirectional long short-term memory architecture combined with regular time-delay neural network layers, all trained with lattice-free MMI [35].

As explained in Section 3, we trained word, morph, and character n-gram models with the VariKN toolkit [17]. For first-pass recognition we created models that have approximately 5–8 million n-gram contexts. For lattice rescoring we created much larger models by tuning the growing and pruning parameters. We stopped training when the models did not increase in size anymore or needed more than 100GB of memory. The morph segmentations were created using the Morfessor toolkit [28, 29]. We trained a few models with different morph vocabulary sizes and selected the best using speech recognition error rate on the development set. For Arabic this resulted in a vocabulary of 17k morphs; in Finnish the selected vocabulary had 30k morphs.

Besides n-gram models we also trained RNNLMs using the TheanoLM toolkit [36]. For all models we used the same basic architecture of a projection layer, an LSTM layer, and a highway layer.

For the Finnish models trained on the full datasets, we used 200 neurons in the projection layer and 1000 neurons in both the LSTM and highway layers. The models trained on only 10% of the data used less neurons, only 400 neurons for both the LSTM and highway layers.

The large Arabic models were already trained for our MGB-3 submission [37]. These models are not completely the same for all types of units, and we did not have enough computing time to make the models equivalent. The word models have larger hidden layers, possibly giving them an advantage over character models and the other subword models. The subword models have 200 neurons in the projection layers and 1000 in the other layers. The hidden layers in the word models are larger, 300 neurons in the projection and 1500 neurons in the LSTM and highway layer. For the models trained on 10% of the language modeling data, we used equal parameters for all units, with 50 neurons in the projection layer and 300 in the hidden layers.

In RNNLMs the computational complexity in the output layer depends on the number of words in the vocabulary. For over 10k word and subword vocabularies, we used classes to reduce the size of the output layer. The words and subwords were grouped into classes using the exchange word clustering algorithm [38, 39]. We used 2000 classes in all word and subword experiments reported here.

During training we used the adaptive gradient (Adagrad) algorithm to update the parameters of RNNLMs. The parameters of the model were updated after processing a mini-batch of training examples. The mini-batch size for character and subword models was 64 and for words 32 sequences. The maximum sequence length depended on the task: 100 for characters, 50 for subwords, and 25 for words. We used an initial learning rate of 0.1 in all experiments. A dropout of 0.2 was used to regularize the parameter learning.

For rescoring we implemented a version of TheanoLM to rescore lattices in the Kaldi format. The different pruning parameters during decoding—beam, recombination order, and the maximum number of tokens per node—were optimized to keep reasonable computation times and follow the memory limits on our computing cluster.

4.3. Results

Table 1a shows the speech recognition results for the Finnish broadcast news set. The results show that the sub-30k model has the lowest word error rate for all the scenarios. In the n-gram results we see that the word model outperforms the character-based model for the full dataset, while for the under-resourced scenario the character-based model improves over the word model by 6%. Comparing the RNNLM results we see that character-based models outperform the word models in both scenarios.

In general the +m+ style markers work best for Finnish, both for the sub-30k and the character model. This is in line with the results in [3].

The results for Arabic in Table 1b show a similar pattern as the Finnish results. The RNNLM-rescored results show that character-based models outperform word models but underperform the sub-17k model. For the 10%-data scenario the difference between word and character-based models is the largest, which confirms the hypothesis that character models have a greater benefit in under-resourced scenarios.

Different marking styles are optimal for Arabic than for Finnish. It is unclear what causes the differences between the markers, and the different optima between languages; we plan to investigate this in future work.

5. PREDICTING UNSEEN WORDS

One of the strengths of subword models is the ability to predict a larger vocabulary of words than any word vocabulary can do. The character models can predict any word and other character sequence, as long as the alphabet matches. To demonstrate how this works in practice, we set up the following experiment to check the number of out-of-corpus (OOC) words that our model predicts correctly. Note that we do not speak of out-of-vocabulary words as the character models have an unlimited vocabulary.

For all words in our test set transcriptions that do not appear in the language model training corpus we look if they appear in the recognized transcription. We use the best transcription after RNNLM rescoring, the same as what was generated in Section 4. For both the character and subword model we use the marking style that performed best in the ASR experiment.

Table 1: Word Error Rates for ASR experiments. Language models are trained on either the full or 10% of the data. <w> is the tagging of word-boundaries with a separate symbol. +m and m+ are the techniques of marking on the left or right of each subword when there is no word-boundary. +m+ marks this, redundantly, on both sides of the subword.

(a) Finnish

		full		10 %	
		n-gram	RNNLM	n-gram	RNNLM
word		**16.70**	**15.04**	**19.63**	**18.29**
sub-30k	+m+	**15.77**	**14.03**	**17.47**	**15.96**
	+m	16.39	14.29	18.60	16.17
	m+	16.26	14.21	18.56	16.40
	<w>	15.95	14.28	17.91	16.57
char	+m+	**16.93**	**14.52**	**18.50**	**17.53**
	+m	16.97	14.69	18.94	17.54
	m+	17.11	14.62	19.22	17.79
	<w>	17.34	14.56	19.09	17.59

(b) Arabic

		full		10 %	
		n-gram	RNNLM	n-gram	RNNLM
word		**17.66**	**16.51**	**19.36**	**18.45**
sub-17k	+m+	**17.15**	16.19	18.43	17.34
	+m	17.40	16.34	18.71	17.47
	m+	**17.15**	**16.02**	**18.17**	**17.24**
	<w>	17.24	16.29	18.38	17.56
char	+m+	18.08	16.90	19.05	18.23
	+m	18.27	16.89	19.12	18.32
	m+	18.26	16.68	19.26	18.28
	<w>	**17.81**	**16.44**	**18.89**	**17.88**

Table 2: The out-of-corpus (OOC) rate for the different dataset and language modeling corpora and proportion of OOCs that were actually recognized correctly by the char, sub-*xx*k and word models.

	model	OOC	char	sub-*xx*k	word
Finnish	full	2.3%	40.9%	37.2%	0%
Finnish	10%	4.7%	47.3%	45.7%	0%
Arabic	full	2.0%	16.3%	17.5%	0%
Arabic	10%	3.5%	31.7%	30.4%	0%

Table 2 shows the percentage of words that were correctly present in the transcription even though they did not appear in the training corpus. Naturally, the word model never predicts any OOC word as it is unable to model words not seen in the training corpus. Both the morph and the character models were able to predict OOC words. In Finnish there is a clear gap between character and sub-30k models, character models recognizing 3.5% more OOC words using the reduced LM corpus and 10% more using the model trained on the full LM corpus. In Arabic the character model performs 7% worse using the full training data and 4% better using the reduced corpus when compared to the sub-17k model.

For many applications, such as keyword spotting, it is not necessary that words are present in the 1-best transcription, but that they are present as likely transcriptions in the search network (lattice). Also, presence of words in the top part of the search network indicate an opportunity for optimization of the language model so that these words would be recognized correctly. Figure 1 shows the proportion of OOC-words in the transcription that are present in the (RNNLM rescored) lattice for Finnish and Arabic.

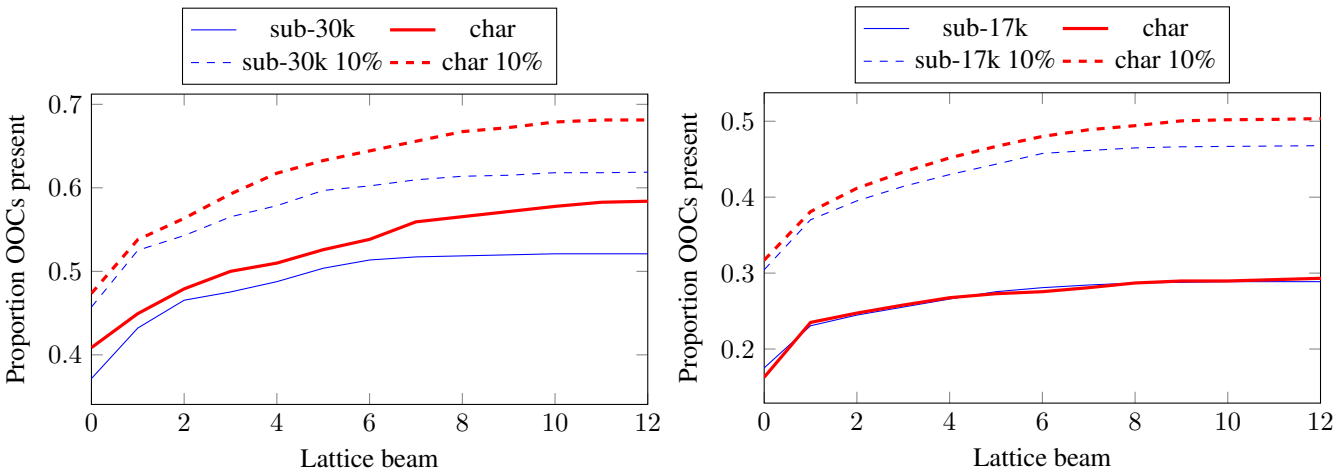
For Finnish the figure shows that character models have a clear benefit over Morfessor-based subword models. The proportion of recognized OOCs is always larger than for the sub-30k model, and the OOC proportion for sub-30k does not increase anymore beyond beam 8. In Arabic the number of recognized OOCs for the full language model data shows an almost identical pattern for both the character and the sub-17k segmentation. The sub-17k subword and character models work equivalently in this scenario. When using the 10% corpus the characters are clearly superior, indicating superior modeling performance in under-resourced scenarios.

6. COMPUTATIONAL REQUIREMENTS OF THE CHARACTER MODEL

As mentioned in Section 3 the computational requirements of processing character and word models are different. We expect character n-gram modeling to be more expensive, because of the very long contexts, and character RNNLM modeling to be cheaper than word modeling, because of the reduced input and output layer sizes.

Table 3 shows the time and memory requirements for the most important steps of the speech recognition training and decoding process. The first part shows the training steps. As expected the largest differences are in the training of the n-gram and RNNLM models. Looking at the small n-gram model training, the memory usage is about 2.5x higher for the character model than for the word model. For training larger n-grams, the difference is smaller. We must note that all models were variable-order models, which is rather memory consuming. This is a necessity for subword models but rather unusual for word models.

In the RNNLM training we see a remarkable difference between words and characters. Whereas for the word model



(a) Finnish.

(b) Arabic.

Fig. 1: Proportion of OOC-words present in beam-pruned lattice.

Table 3: Time and memory usage for the training and decoding of the Finnish datasets (full)

Phase	**char**		**word**	
	time	**mem.**	**time**	**mem.**
n-gram training small	3h	11G	1h	4G
n-gram training	19h	90G	16h	70G
RNNLM training	48h	28M	116h	2G
L.fst creation	<1s	<2G	4m	<2G
G.fst creation	40s	<2G	40s	12G
HCLG combination	2:30m	2.5G	3:50m	3.6G
Decoding	7h	0.6 G	11h	1.3G
Rescoring n-gram	17m	1.8G	13m	1.7G
Rescoring RNNLM	7h	5G	8h	15G

2GB of GPU memory is required, the character model only needs 28MB. Also the number of hours per epoch is much lower and we noticed a faster convergence.

To our surprise, even normal n-gram decoding is much more efficient for characters than for words, even though the resulting lattices are larger. Most likely the decoding FST is more efficient in the character case. Rescoring with n-grams is a bit more expensive for characters, but RNNLM rescoring is again a bit faster. The largest gain there is the memory usage. The reason for similar decoding times is that for character models, the lattices contain more nodes to search through.

The numbers for subword models (not shown) are in between of word and character training.

7. CONCLUSION

We have implemented and evaluated character-based modeling in a state-of-the-art speech recognition systems for Finnish and Arabic. This system outperforms word-based modeling in most scenarios and is clearly better in under-resourced scenarios. Compared to modeling based on statistical morphs, we did not see direct improvement in recognition performance. Furthermore, we have evaluated how the character models predict words that have not been seen in the training data and observed a clear improvement over other subword and word models. When looking at the computational requirements of different models we conclude that character models can give real speed benefits compared to word models, especially in RNNLM training where much lower amount of memory and time per epoch is required.

Overall, the character models can provide benefits compared to word models. Besides the possible improvements in accuracy, the RNNLM can be trained in a much more efficient manner, and the overall system resources requirement is lower than for word-based models.

Although we have not experimented with it yet, we believe that using characters as units can inspire different RNNLM architectures that are not possible with longer units. This is because the majority of the parameters can be put into the inner layers instead of managing high-dimensional input and output layers.

In future we plan to extend this work to other languages, possibly even languages which have less phonemic alphabet than Finnish and Arabic. We will also explore new language model adaptation scenarios that become possible for unlimited vocabulary speech recognition.

8. REFERENCES

[1] Teemu Hirsimäki, Mathias Creutz, Vesa Siivola, Mikko Kurimo, Sami Virpioja, and Janne Pylkkönen, "Unlimited vocabulary speech recognition with morph language models applied to Finnish," *Computer Speech & Language*, vol. 20, no. 4, pp. 515–541, Oct. 2006.

[2] Seppo Enarvi, Peter Smit, Sami Virpioja, and Mikko Kurimo, "Automatic speech recognition with very large conversational finnish and estonian vocabularies," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 11, pp. 2085–2097, 2017.

[3] Peter Smit, Sami Virpioja, and Mikko Kurimo, "Improved subword modeling for WFST-based speech recognition," in *INTERSPEECH 2017 – 18th Annual Conference of the International Speech Communication Association*, Stockholm, Sweden, August 2017.

[4] Florian Eyben, Martin Wllmer, Björn Schuller, and Alex Graves, "From speech to letters - using a novel neural network architecture for grapheme based asr," in *ASRU 2009 – IEEE Workshop on Automatic Speech Recognition & Understanding*, 2009, pp. 376–380.

[5] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31st International Conference on Machine Learning*, Eric P. Xing and Tony Jebara, Eds., Bejing, China, 22–24 Jun 2014, vol. 32 of *Proceedings of Machine Learning Research*, pp. 1764–1772, PMLR.

[6] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The kaldi speech recognition toolkit," in *ASRU 2011 – IEEE Workshop on Automatic Speech Recognition & Understanding*, 2011.

[7] Ahmed Ali, Stephan Vogel, and Steve Renals, "Speech recognition challenge in the wild: Arabic MGB-3," in *ASRU 2017 – IEEE Workshop on Automatic Speech Recognition & Understanding*, December 2017.

[8] Sherry Perdue Casali, Beverly H. Williges, and Robert D. Dryden, "Effects of recognition accuracy and vocabulary size of a speech recognition system on task performance and user acceptance," *Human Factors*, vol. 32, no. 2, pp. 183–196, 1990.

[9] Steve Young, "A Review of Large-vocabulary Continuous-speech Recognition," *IEEE Signal Processing Magazine*, vol. 13, no. 5, pp. 45–57, Sept 1996.

[10] Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai, "Class-based n-gram models of natural language," *Comput. Linguist.*, vol. 18, no. 4, pp. 467–479, Dec. 1992.

[11] Maximilian Bisani and Hermann Ney, "Open vocabulary speech recognition with flat hybrid models," in *INTERSPEECH 2005 – Eurospeech, 9th European Conference on Speech Communication and Technology*, Lisbon, Portugal, September 2005, pp. 725–728.

[12] Joshua Goodman, "Classes for fast maximum entropy training," in *ICASSP 2001 – IEEE International Conference on Acoustics, Speech and Signal Processing*, 2001, vol. 1, pp. 561–564 vol.1.

[13] Hai-Son Le, Ilya Oparin, Abdel Messaoudi, Alexandre Allauzen, Jean-Luc Gauvain, and François Yvon, "Large vocabulary SOUL neural network language models," in *ICASSP 2011 – IEEE International Conference on Acoustics, Speech and Signal Processing*, 2011, pp. 1469–1472.

[14] Frederic Morin and Yoshua Bengio, "Hierarchical probabilistic neural network language model," 2005, pp. 246–252, Society for Artificial Intelligence and Statistics, (Available electronically at http://www.gatsby.ucl.ac.uk/aistats/).

[15] Manhung Siu and Mari Ostendorf, "Variable n-grams and extensions for conversational speech language modeling," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 8, no. 1, pp. 63–75, 2000.

[16] Teemu Hirsimäki, Janne Pylkkönen, and Mikko Kurimo, "Importance of high-order n-gram models in morph-based speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 4, pp. 724–732, May 2009.

[17] Vesa Siivola, Teemu Hirsimäki, and Sami Virpioja, "On growing and pruning Kneser-Ney smoothed n-gram models," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 15, no. 5, pp. 1617–1624, 2007.

[18] Antti Puurula and Mikko Kurimo, "Vocabulary decomposition for estonian open vocabulary speech recognition," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. 2007, pp. 89–95, Association for Computational Linguistics.

[19] Amr El-Desoky Mousa, M. Ali Basha Shaik, Ralf Schlüter, and Hermann Ney, "Morpheme level feature-based language models for German LVCSR," in *INTERSPEECH 2012 – 13th Annual Conference of the International Speech Communication Association*, Portland, OR, USA, September 2012, pp. 170–173.

[20] Andrew Maas, Ziang Xie, Dan Jurafsky, and Andrew Ng, "Lexicon-free conversational speech recognition with neural networks," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2015, pp. 345–354, Association for Computational Linguistics.

[21] Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W. Black, "Character-based neural machine translation," *CoRR*, vol. abs/1511.04586, 2015.

[22] Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio, "A character-level decoder without explicit segmentation for neural machine translation," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2016, pp. 1693–1703, Association for Computational Linguistics.

[23] Tomas Mikolov, Martin Karafit, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH 2010 – 11th Annual Conference of the International Speech Communication Association*, Makuhari, Japan, September 2010, pp. 1045–1048.

[24] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney, "LSTM neural networks for language modeling," in *INTERSPEECH 2012 – 13th Annual Conference of the International Speech Communication Association*, Portland, OR, USA, September 2012, pp. 194–197.

[25] Ghinwa Choueiter, Daniel Povey, Stanley F. Chen, and Geoffrey Zweig, "Morpheme-based language modeling for Arabic LVCSR," in *ICASSP 2006 – IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2006, pp. 1053–1056.

[26] Mathias Creutz, Teemu Hirsimäki, Mikko Kurimo, Antti Puurula, Janne Pylkkönen, Vesa Siivola, Matti Varjokallio, Ebru Arisoy, Murat Saraçlar, and Andreas Stolcke, "Morph-based speech recognition and modeling of out-of-vocabulary words across languages," *ACM Trans. Speech Lang. Process.*, vol. 5, no. 1, pp. 3:1–3:29, Dec. 2007.

[27] Katrin Kirchhoff, Dimitra Vergyri, Jeff Bilmes, Kevin Duh, and Andreas Stolcke, "Morphology-based language modeling for conversational Arabic speech recognition," *Computer Speech & Language*, vol. 20, no. 4, pp. 589 – 608, 2006.

[28] Mathias Creutz and Krista Lagus, "Unsupervised discovery of morphemes," in *Proceedings of the ACL 2002 Workshop on Morphological and Phonological Learning*, Stroudsburg, PA, USA, 2002, vol. 6 of *MPL '02*, pp. 21–30, Association for Computational Linguistics.

[29] Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo, "Morfessor 2.0: Python implementation and extensions for Morfessor Baseline," Report 25/2013 in Aalto University publication series SCIENCE + TECHNOLOGY, Department of Signal Processing and Acoustics, Aalto University, Helsinki, Finland, 2013.

[30] Dorota J Iskra, Beate Grosskopf, Krzysztof Marasek, Henk van den Heuvel, Frank Diehl, and Andreas Kiessling, "SPEECON-Speech databases for consumer devices: Database specification and validation.," in *LREC*, 2002.

[31] Antti Rosti, Anssi Rämö, Teemu Saarelainen, and Jari Yli-Hietanen, "Speechdat Finnish database for the fixed telephone network," Tech. Rep., Tampere University of Technology, 1998.

[32] André Mansikkaniemi, Peter Smit, and Mikko Kurimo, "Automatic construction of the Finnish Parliament Speech Corpus," in *INTERSPEECH 2017 – 18th Annual Conference of the International Speech Communication Association*, Stockholm, Sweden, August 2017.

[33] Ahmed Ali, Peter Bell, James Glass, Yacine Messaoui, Hamdy Mubarak, Steve Renals, and Yifan Zhang, "The MGB-2 challenge: Arabic multi-dialect broadcast media recognition," in *SLT 2016 – IEEE Spoken Language Technology Workshop*, December 2016, pp. 279–284.

[34] CSC - IT Center for Science, "The Helsinki Korp Version of the Finnish Text Collection," 1998.

[35] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi," in *INTERSPEECH 2016 – 17th Annual Conference of the International Speech Communication Association*, San Francisco, September 2016, pp. 2751–2755.

[36] Seppo Enarvi and Mikko Kurimo, "TheanoLM an extensible toolkit for neural network language modeling," in *INTERSPEECH 2016 – 17th Annual Conference of the International Speech Communication Association*, San Francisco, September 2016, pp. 3052–3056.

[37] Peter Smit, Siva Reddy Gangireddy, Seppo Enarvi, Sami Virpioja, and Mikko Kurimo, "Aalto system for the 2017 Arabic multi-genre broadcast challenge," in *ASRU 2017 – IEEE Workshop on Automatic Speech Recognition & Understanding*, December 2017.

[38] Sven Martin, Jörg Liermann, and Hermann Ney, "Algorithms for bigram and trigram word clustering," *Speech communication*, vol. 24, no. 1, pp. 19–37, 1998.

[39] Rami Botros, Kazuki Irie, Martin Sundermeyer, and Hermann Ney, "On efficient training of word classes and their application to recurrent neural network language models.," in *INTERSPEECH 2015 – 16th Annual Conference of the International Speech Communication Association*, Dresden, Germany, September 2015, pp. 1443–1447.