Character-based units for Unlimited Vocabulary Continuous Speech Recognition

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

Aalto University
School of Electrical
Engineering

Department of Signal Processing and Acoustics, Aalto University, Finland

What are the best LM units?

In ASR words are most common.

- **Words**: the vocabulary limits the language, data sparseness limits the model performance, very large vocabulary increases the training and recognition time.

But for some system•s sub•word•s excel.

- **Subwords**: a small set of subwords can model and generate words not seen in the training data, linguistic or statistical tools needed to define the units, the choice of optimal units needs experimentation, LMs require longer contexts.

W•i•l•l c•h•a•r•a•c•t•e•r s•y•s•t•e•m•s w•o•r•k ?

- **Characters**: this paper analyzes the performance with respect to the training resources, recognition performance and ability to model unseen words.

Character-based Language Models

- Words and sentences are made of long sequences of characters.
- *n*-gram techniques like varigram growing and pruning can model long sequences, but the current tools are not optimal for very long sequences.
- Recurrent neural network language models (RNNLM) can model long sequences, but very large vocabularies require very large input and output layers.
- A small subword vocabulary reduces data sparsity, because short units occur frequently.
- A character LM is easy to adapt, because even though some words are rare, most character combinations are not. Thus, the LM probability of rare or unseen words can be adapted without seeing them.

Implementation

- We built TDNN-BLSTM acoustic models with **Kaldi** with >1000 hours of training data per language.
- The phonemesets are **grapheme-based**. The Lexicon FST is modified to model boundaries between subwords/characters correctly. (Smit et al., Interspeech 2017)
- The subword models are created with **Morfessor** and are optimized for each language.
- We use variable order ***n*-gram** models, trained with the **VariKN** toolkit.
- We train **RNNLM** models with **TheanoLM**.
- We test on **Finnish** broadcast news (5.4 hours, 127 speakers) and **Arabic** MGB-2 development data (8.4 hours, 107 speakers). More Arabic results are available in our MGB-challenge paper.
- For Finnish we train the LM's on newspaper texts (156M training tokens). For Arabic we use the MGB-2 text corpus (125M training tokens).
- We also train language models on 10% of the original corpora, to simulate a lower-resource scenario.

Finnish

	full		**10 %** (15.6M)	
	***n*-gram**	**RNNLM**	***n*-gram**	**RNNLM**
word	16.7	15.0	19.6	18.3
subword	15.8	14.0	17.5	16.0
char	16.9	14.5	18.5	17.5

Arabic

	full		**10 %** (12.5M)	
	***n*-gram**	**RNNLM**	***n*-gram**	**RNNLM**
word	17.7	16.5	19.6	18.5
subword	17.2	16.0	18.2	17.2
char	17.8	16.4	18.9	17.9

% Word Error Rate

Effect on Out-of-vocabulary

- We know that theoretically all words can be predicted in subword and character models. We designed an experiment to see if this happens in practice.
- In a recognition task we look to all words in the transcription that were not present in the language modeling corpus, out-of-corpus (**OOC**) words. For word models the OOC rate is equivalent to the out-of-vocabulary rate (**OOV**).
- We report the the **recovery rate**, the amount of OOC tokens that were actually present in the recognition output.
- Besides the 1-best output, we also tested the recovery rate in the complete lattice, for different lattice beams.

Results

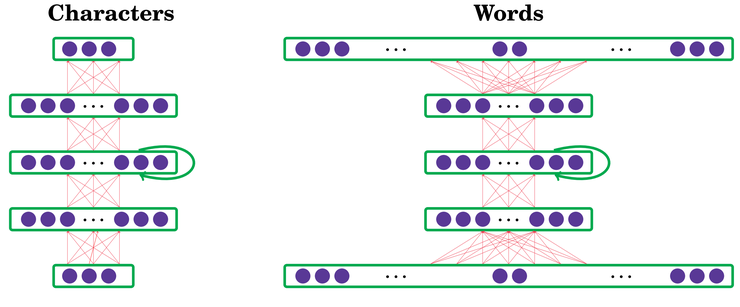
- As expected, character models are stronger in recognizing out-of-vocabulary words. Also, they perform better then subword models.
- The effect is even stronger when using to output lattices. For the Finnish 10% model the character model has 65% of the OOC-words present in the recognition-lattice with beam 6. More results are available in the paper.

			Recovery rate		
model		**OOC**	**char**	**subword**	**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%

OOC rate equals proportion of tokens in evaluation set not in the training data. Recovery rate is the proportion of OOC tokens present in the speech recognition output.

How would you like your neural networks to look like?

- A very large vocabulary makes the RNNLM input and output layer very large: 1 M word input and 1000-dim first hidden layer take 1 B parameters!
- The data sparseness makes this hard to train. If the input and output dimensions are reduced, a lot more parameters can be spent on making the model deeper.



System requirements

Compared to word models, some parts of training and decoding will take different amount of resources.

_n_-gram Model training

The character *n*-gram LMs need higher-order *n*-grams , which takes more time and space to compute.

	time	memory
char	3h	11G
word	1h	4G

Neural Network Model training

In word-based RNNLMs the input and output layer need lots of parameters, which need more memory and time to train.

	time	memory
char	48h	28M
word	116h	2G

Decoding and rescoring

During decoding and RNNLM rescoring the larger word vocabulary requires more memory, but the differences are smaller than in training.

Decoding	time	memory
char	7h	0.6G
word	11h	1.3G

Rescoring	time	memory
char	7h	5G
word	8h	15G

Conclusions

- It is possible to build, with conventional tools, a successful speech recognition system that is build on characters instead of words.
- Character models can outperform word models, especially in case of smaller amounts of text resources.
- Character models are better in predicting unseen words then subword models with larger units.
- In RNNLM training character units are faster and easier to train then subword or word models.

Future work

- Comparison with end-to-end systems that are also character-based.
- Expansion to nonphonemic languages such as English.
- Experimenting with different RNNLM architectures.
- Adaptation for character-based language models.