Acta Linguistica Academica 69 (2022) 4, 581–598
DOI: 10.1556/2062.2022.00582

AKADÉMIAI KIADÓ

Morphology aware data augmentation with neural language models for online hybrid ASR

BALÁZS TARJÁN[1,2*], TIBOR FEGYÓ[1,2] and PÉTER MIHAJLIK[1,3]

[1] Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Budapest, Hungary

[2] SpeechTex Ltd., Budapest, Hungary

[3] THINKTech Research Center, Budapest, Hungary

Received: April 30, 2022 • Accepted: October 2, 2022
Published online: November 21, 2022

ABSTRACT

Recognition of Hungarian conversational telephone speech is challenging due to the informal style and morphological richness of the language. Neural Network Language Models (NNLMs) can provide remedy for the high perplexity of the task; however, their high complexity makes them very difficult to apply in the first (single) pass of an online system. Recent studies showed that a considerable part of the knowledge of NNLMs can be transferred to traditional n-grams by using neural text generation based data augmentation. Data augmentation with NNLMs works well for isolating languages; however, we show that it causes a vocabulary explosion in a morphologically rich language. Therefore, we propose a new, morphology aware neural text augmentation method, where we retokenize the generated text into statistically derived subwords. We compare the performance of word-based and subword-based data augmentation techniques with recurrent and Transformer language models and show that subword-based methods can significantly improve the Word Error Rate (WER) while greatly reducing vocabulary size and memory requirements. Combining subword-based modeling and neural language model-based data augmentation, we were able to achieve 11% relative WER reduction and preserve real-time operation of our conversational telephone speech recognition system. Finally, we also demonstrate that subword-based neural text augmentation outperforms the word-based approach not only in terms of overall WER but also in recognition of Out-of-Vocabulary (OOV) words.

KEYWORDS

speech recognition, data augmentation, RNNLM, Transformer, conversational telephone speech, morphologically rich language

[*] Corresponding author. E-mail: tarjanb@tmit.bme.hu

1. INTRODUCTION

Today's state-of-the-art in language modeling for ASR relies on Neural Network Language Models (NNLMs) (Mikolov et al. 2010; Sundermeyer et al. 2012; Irie et al. 2019), capable of handling continuous space and thereby outperforming traditional Back-off N-gram LMs (BNLMs). BNLMs cannot exploit long context based syntactic dependencies and are also less flexible in terms of generalization for unseen cases, as semantic knowledge (such as embeddings reflecting similarity) is not captured while training them.

NNLMs however have an undesired property, they are computationally very heavy in decoding, so neural LMs cannot be effectively used in a single decoding pass, they are rather exploited by rescoring lattices obtained from a first decoding pass with a BNLM. It is obvious, but can also be shown, that information is lost during the first decoding pass, as the pruning of the recognition network is based only on short context syntax, discarding both longer context syntactic and quasi all semantic knowledge. Another problem arising is the increased latency of the system through the two decoding passes, which hampers exploitation in strict online requirements.

To reduce these limitations in exploiting neural LMs for ASR, several solutions have been proposed (Deoras et al. 2011; Arisoy et al. 2014; Adel et al. 2014; Singh et al. 2019). In Adel et al. (2014) it was shown that using the neural LM to generate an augmented training corpus to train an improved BNLM is the best performing strategy. Sometimes these are called approximative models as they try to capture the knowledge of the neural model through their augmented training corpus. Although the converted model loses its ability to model long contexts and distributed input features, it can be directly applied for first-pass decoding that makes these techniques attractive.

Another burden of language modeling for morphologically rich languages are the different syntactic properties of the language compared to English. Heavy agglutination results in much larger vocabularies, which is a problem in itself, but causes other problems too: individual word forms occur less often and hence, the size of the training corpus should accordingly be augmented to maintain the predictive power of the dataset. Moreover, as suffixes express grammatical relations usually provided by word order in English, morphologically rich languages tend to be more permissive in choosing word order, leading to higher variation. This impairs BNLM estimation badly, but may also cause that word embeddings become less powerful in terms of syntactic and semantic consistency (Döbrössy et al. 2019), even despite using long context windows. To alleviate these problems linked to the different organization of morphologically rich languages, subword unit modeling is an often used alternative (Creutz & Lagus 2002; Kurimo et al. 2007; Mihajlik et al. 2010).

1.1. Related work

Recently several studies concentrated on the approximation of neural LMs. Suzuki et al. (2019) uses a domain balanced mixture of the training corpora to train a shallow Recurrent Neural Network Language Model (RNNLM) for text generation, and improve speech recognition results for Japanese, Korean and English. They use subword-based approach, but compose these subwords back into words to prepare the final LM, unlike our approach that retokenizes words into subword units in the final LM. Another approach called RNN n-gram has also been introduced (Chelba et al. 2017). RNN n-grams are special RNNLMs trained on n-grams sampled from the training data. As a consequence, the size of the modeled context here is also limited, but RNN n-gram models are able to learn word embeddings just like standard RNNLMs. Wang et al. (2019) report using general domain pre-trained Transformer (Vaswani et al. 2017) to generate augmentation text corpora for LM training. They demonstrate that the pre-trained and fine-tuned Transformer performs significantly better in data augmentation than RNNLM or simple in-domain Transformer models.

Although subword language modeling has been used in morphologically rich Finnish ASR systems for more than a decade now (Creutz & Lagus 2002; Kurimo et al. 2007), it was not found beneficial for spontaneous conversational speech until recently. In Enarvi et al. (2017), subword RNNLMs were trained on Finnish and Estonian conversations and used for rescoring lattices generated with conventional back-off models. In a recent paper (Singh et al. 2019), n-gram based approximation of recurrent language models was evaluated on a Finnish and an Arabic OOV keyword retrieval task. The approximation method significantly improved OOV search results, however the proposed model was not tested on in-vocabulary words and no overall word error rates were presented either.

Subword language models have already been applied successfully for recognition of Hungarian conversational speech (Mihajlik et al. 2010; Tarján et al. 2013), but subword-based neural language models have not been used before to the best of our knowledge except for our former studies in the topic (Tarján et al. 2019, 2020a, b). In the first paper (Tarján et al. 2019), the performance of BNLMs, and models augmented with RNNLMs were compared in terms of perplexity and WER on a Hungarian conversational telephone speech recognition task. In our next paper (Tarján et al. 2020b), we analyzed the effectiveness of the RNNLM based data augmentation by systematically comparing BNLMs, augmented models and 2-pass recognition results to determine the amount of knowledge that can be transferred from the offline to the online ASR system. Our first Transformer based results were presented in Tarján et al. (2020a), where we showed that language models augmented with a pretrained Transformer LM can significantly outperform models augmented with an RNNLM. In addition, the first version of our subword-based data augmentation approach was also introduced in that paper.

1.2. Our contribution

This paper summarizes the results of our research aimed at gaining a better understanding of how to transfer the knowledge of modern neural network based language models to the conventional back-off n-gram LMs. Our goal is to improve the LM of an online call center ASR system in the morphologically rich Hungarian. We compare conventional BNLMs and n-gram approximation of RNNLMs, non-pretrained and pre-trained Transformer LMs. With the neural language models we generate training text for a BNLM and demonstrate that such data augmentation is efficient in Hungarian, if vocabulary is large enough and a large BNLM is used.

We also propose a morphology aware data augmentation method by retokenizing the augmented training corpus to subword units, and training a subword-based BNLM on it. We demonstrate that (i) the ASR accuracy further improves compared to the word based baseline augmented BNLM, and (ii) the footprint and complexity of the resulting subword unit augmented BNLM significantly decrease. As subword unit LMs are known to perform better on a wide range of morphologically rich languages, we hypothesize that our approach is transferable to other such languages.

In addition to summarizing our former studies, our paper also extends them in several aspects: (i) recurrent and Transformer models are discussed in parallel, this way providing a much better comparison among these techniques; (ii) the morphology aware Transformer based data augmentation process has been greatly revised to make the process much clearer and more efficient compared to the method presented in Tarján et al. (2020a); (iii) we introduce new experimental results about the impact of vocabulary and LM size on the accuracy of augmented models to illustrate the drawbacks of word-based data augmentation for morphologically rich languages; (iv) we compare the OOV recognition capabilities of the discussed language modeling techniques; (v) finally, we apply the recurrent and Transformer based data augmentation simultaneously and prove that these neural LM techniques can support each other.

In the next section the experimental database and preprocessing methods are introduced. In Section 3, we describe the techniques we used for training our different types of language models. Next, Section 4 introduces the data augmentation techniques applied in our experiments. In Section 5 the speech recognition results are presented. Finally, Section 6 highlights the most impactful outcome of our work.

2. DATABASE

2.1. Training data

2.1.1. Original data. In-domain training data is extracted from the Hungarian Call Center Speech Database (HCCSD) consisting of anonymised telephone customer service calls and the corresponding manual transcripts. We selected 290 h of recordings for training the acoustic model of our ASR system (see Table 1). The in-domain LMs are built on the transcripts of the training set containing 3.4M word tokens and 100k unique word forms. As the available in-domain training text data is very limited, we also utilize a general text corpus for pre-training the Transformer LM, which was collected from the website of the Hungarian National Assembly[1] and contains official transcripts of parliamentary speeches.

Table 1. Train and test dataset statistics

In-domain	Train	Validation	Evaluation
Audio [h:m]	290:07	7:31	12:12
# of word tokens	3,401,775	45,773	66,312
word OOV rate [%]	–	2.7	2.5
General text			
# of word tokens	57,601,277	–	–

[1] www.parlament.hu.

2.1.2. Subword segmented data. Morphologically rich languages have significantly larger vocabulary, as case endings usually reflect grammatical roles. Large vocabulary size can be a problem in itself, however it also increases data sparseness in the training data and result high OOV rate. A common remedy is to segment words into smaller units and train language models on these subword sequences (Kurimo et al. 2007; Mihajlik et al. 2010). One of the most popular statistical word segmentation algorithm is Morfessor (Creutz & Lagus 2002) was inspired by the Minimum Description Length (MDL) principle, and was specifically designed for processing morphologically rich languages. We apply the Python implementation of the original algorithm called Morfessor 2.0 (Virpioja et al. 2013). Hyperparameters of the segmentation were optimized on the validation test set (see Section 2.2).

Non-initial morphs of every word were tagged with the '+' sign to provide information to the ASR decoder for the reconstruction of word boundaries (see left-marked style in Smit, Virpioja & Kurimo 2017). For instance, subword segmentation transcribes the Hungarian sentence '*megbeszélem a nejemmel*' (meaning 'I will discuss it with my wife') as follows:

Conventional tokenization:
hát megbeszélem a nejemmel
Subword-based tokenization:
hát meg +beszél +em a nejem +mel

2.2. Test data

Almost 20 h of conversations were selected from HCCSD for testing purposes. The test dataset was split into two disjoint parts (see Table 1). The validation set (≈7.5 h) and the corresponding text transcripts were used for optimization of the hyperparameters (e.g. learning rate control, early stopping), whereas evaluation set (≈12 h) was used to test the models and report experimental results. Subword segmentation of evaluation dataset was performed with Morfessor 2.0 toolkit using the segmentation model we optimized on the validation set.

3. LANGUAGE MODELING

3.1. Back-off n-gram models

Count-based, back-off language models (BNLMs) have low computational cost and fit well into Weighted Finite-State Transducer (WFST) framework, hence are still widely used in online, single-pass ASR systems. We carry out training and interpolation of BNLMs with the SRI language modeling toolkit (Stolcke 2002) applying Chen and Goodman's modified Kneser-Ney discounting (Chen & Goodman 1999).

3.2. Recurrent language model

The 2-layered Long Short-Term Memory (LSTM) RNNLM structure (Elman 1990; Hochreiter & Schmidhuber 1997) we used in our experiments is illustrated in Figure 1. This type of network has already been successfully applied for other language modeling tasks (Zaremba et al. 2014;

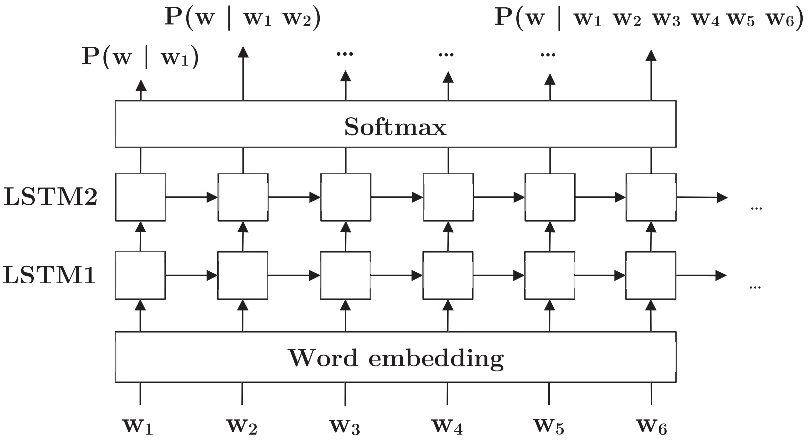

Figure 1. The recurrent LSTM language model structure used in our experiments

Chelba et al. 2017). Our implementation[2] is based on the TensorFlow sample code of the Penn Tree Bank language model presented in Zaremba et al. (2014).

The hyperparameters of the neural network were optimized on the validation set. One batch consists of 32 sequences containing 35 tokens each (words or subwords). LSTM states are preserved between the batches, so *stateful* recurrent networks are trained according to TensorFlow terminology. The 650 dimension word/subword embedding vectors are trained on the input data, since we did not find any benefit of Hungarian pretrained embeddings. In order to match the dimensionality of embeddings the size of the LSTM layer is also set to 650. The applied recurrent LSTM structure uses altogether 20 million trainable parameters. After testing several optimization algorithms, we decided on the momentum accelerated, Stochastic Gradient Descent (SGD). We apply a simple adaptive learning rate decay scheme, where the initial learning rate is set to 1 and decreased by a factor of 2 each time validation error plateau. For regularization purposes, dropout layers with keep probability of 0.5 and early stopping with patience of 3 epochs are used.

3.3. Transformer language model

Recently Transformer architectures have proven to be particularly successful in generating well-structured, high-quality texts thanks to the self attention mechanism and the depth of the model (Radford et al. 2019; Yang et al. 2019). In order to generate augmentation text to our ASR task, we applied one of the most promising Transformer architectures called OpenAI GPT-2 (Radford et al. 2019) implemented in HuggingFace's Transformers library (Wolf et al. 2019). The GPT-2 architecture has four variants with different sizes from which we opted for the *medium* having altogether 345 million trainable parameters.

[2] github.com/btarjan/stateful-LSTM-LM.

GPT-2 *medium* consists of 24 decoder-only Transformer blocks each having 16 attention heads and 1,024 dimensional embedding and bottleneck layers (see Figure 2). Due to GPT-2 conventions the first feed-forward layer in each block is four times larger than the bottleneck layers (4,096). For regularization purposes it applies embedding, attention and residual dropouts with a rate of 0.1. We apply the Adam optimization scheme (Kingma & Ba 2014) with initial learning rate of 1e-4 and a linear decay schedule. We pre-train the model on the general training corpus for 15 epochs using minibatches of 16 sequences consisting of 512 tokens each. Fine-tuning of the pre-trained model took 4 epochs on the in-domain training set with the same hyperparameters as in pre-training. Tokenization was performed with a byte-level Byte Pair Encoding (BPE) (Sennrich et al. 2015) model with 30k vocabulary items (256 bytes +29,744 merges) trained on the in-domain training set.

In addition to the pre-trained model, we also trained a model exclusively on the in-domain training text. Despite the small amount of training data, we did not encounter any convergence problems during the 10 epochs of training. The relative robustness of this non-pretrained Transformer model may be surprising; however, our results are in line with the recently developed theory of overparameterized machine learning (Dar et al. 2021).

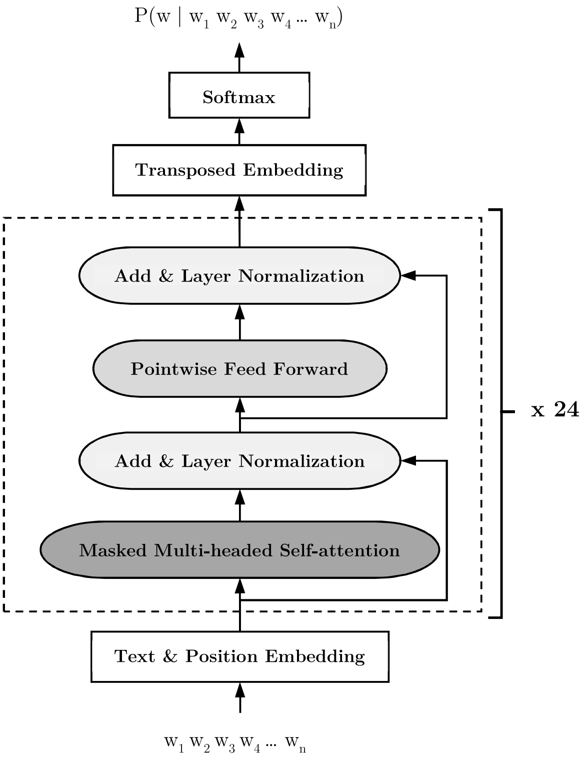
P(w | w1 w2 w3 w4 ... wn)
Softmax
Transposed Embedding
Add & Layer Normalization
Pointwise Feed Forward
Add & Layer Normalization
Masked Multi-headed Self-attention
x 24
Text & Position Embedding
w1 w2 w3 w4 ... wn

Figure 2. Structure of the applied GPT-2 *medium* architecture

4. DATA AUGMENTATION WITH NEURAL LANGUAGE MODELS

There are various approaches for the approximation of a neural language model with a back-off, n-gram language model (Deoras et al. 2011; Arisoy et al. 2014; Adel et al. 2014; Singh et al. 2019). In Adel et al. (2014) three such methods are described and evaluated, coming to a conclusion that the so called text generation based data augmentation yields the best results. The main idea of this approach is to estimate the BNLM parameters from a large text corpus generated by a NNLM. In this section, we describe the various data augmentation approaches applied in our experiments.

4.1. Data augmentation with RNNLM

We first generate a large text corpus with the RNNLM, which is than can be used to train a backoff, n-gram language model (see Figure 3). The model trained on the generated text can be considered as the n-gram approximation of the recurrent neural model (RNN-BNLM). To further improve the performance, the RNN-BNLM can be interpolated with the original, in-domain BNLM (BNLM + RNN-BNLM) and can be utilized in a real-time ASR system. Interpolation weights are optimized on the development set. In our work, we first generated 100 million words/subwords with the corresponding word or subword-based RNNLM (RNN-BNLM 100M) that was formerly trained on the in-domain training set. In order to get an insight how the corpus size influences the language model capabilities, we also generated a larger text corpus with 1 billion subwords (RNN-BNLM 1B).

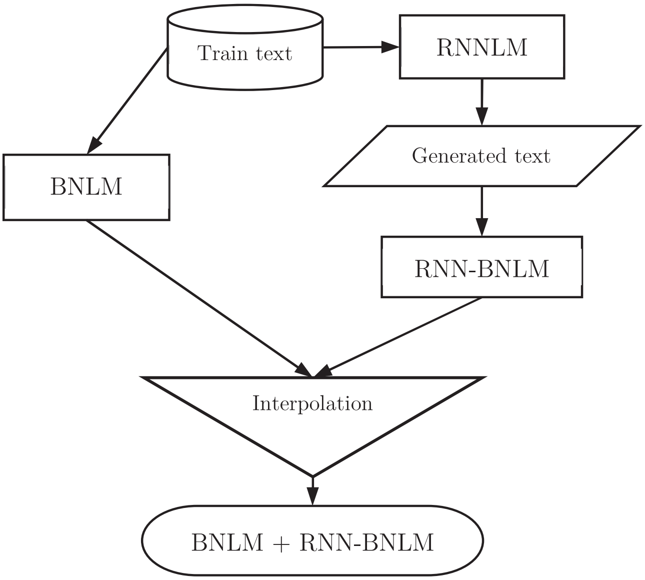

Figure 3. Data augmentation of in-domain training text with recurrent neural language model

4.2. Data augmentation with Transformer LM

4.2.1. Neural text generation with GPT-2. Generation of a text sequence is initialized with a prefix prompt, which we sample from the in-domain training set. The length of the sampled prefix varies randomly between 1.7 words to balance the trade-off between free and constrained text generation. For the same reasons, the temperature is also randomly changed from 1.0 to 1.5. We generate two large corpora for data augmentation purposes each consisting of 1 billion words. The first corpus is generated with the pre-trained and then fine-tuned Transformer LM (TR) described in Section 3.3, while the second one is generated with a Transformer trained directly on the in-domain corpus without pre-training (TR-noPT).

4.2.2. Word-based data augmentation. The fact that text corpora generated by RNNLMs can improve the accuracy of n-gram language models has been shown by several studies before (Deoras et al. 2011; Adel et al. 2014; Suzuki et al. 2019; Tarján et al. 2019, 2020b). However, Wang and her colleagues (Wang et al. 2019) were the first who applied general domain pre-training and in-domain fine-tuning of a Transformer LM to improve the effectiveness of the data augmentation process. For that reason we summarize their original, word-based data augmentation process in this section. In the next section, we are going to propose an extended version of the augmentation process that fits better to morphologically rich scenarios.

The original, word-based version of neural text based data augmentation process is shown on the left side of Figure 4 (white boxes). First a large corpus is generated by the Transformer LM (pre-trained on a general text corpus and fine-tuned on the in-domain text). Based on this generated text a BNLM (TR-BNLM) is trained, which approximates the short-term dependencies learned by the Transformer. To further improve the model, the TR-BNLM can be interpolated with a BNLM trained the on the in-domain text (BNLM + TR-BNLM).

4.2.3. The proposed subword-based data augmentation. Language modeling of morphologically rich languages poses a great challenge, since the large number of word forms cause data sparseness and high OOV rate. A common remedy is to segment words into smaller parts and train language models on these subword sequences (Kurimo et al. 2007; Mihajlik et al. 2010). In this study we compare two popular, data-driven subword tokenizers Morfessor (Creutz & Lagus 2002) and BPE algorithm (Sennrich et al. 2015) for the character-level retokenization of the generated corpora. Morfessor is inspired by the Minimum Description Length (MDL) principle and specifically designed for processing morphologically rich languages (see Section 2.1). BPE is one of the most widely used subword tokenizers especially for processing byte-level token sequences (Radford et al. 2019).

The question arises as to why to retokenize the generated texts when we could use the original byte-level tokenizer of the Transformer model. Hungarian uses many characters that are encoded with multiple bytes. A byte-level tokenizer can split these multibyte characters and place the leading and trailing part of a character into separate subwords. These broken characters make assignment of phonetic transcripts ambiguous, since subword boundaries does not represent phone boundaries anymore. By using character-level retokenization we can prevent this ambiguity in the phonetic transcription of subwords.

Our proposed morphology aware extension to the word-based data augmentation process called *subword-based neural text augmentation* is depicted in Figure 4 (grey boxes). The revised

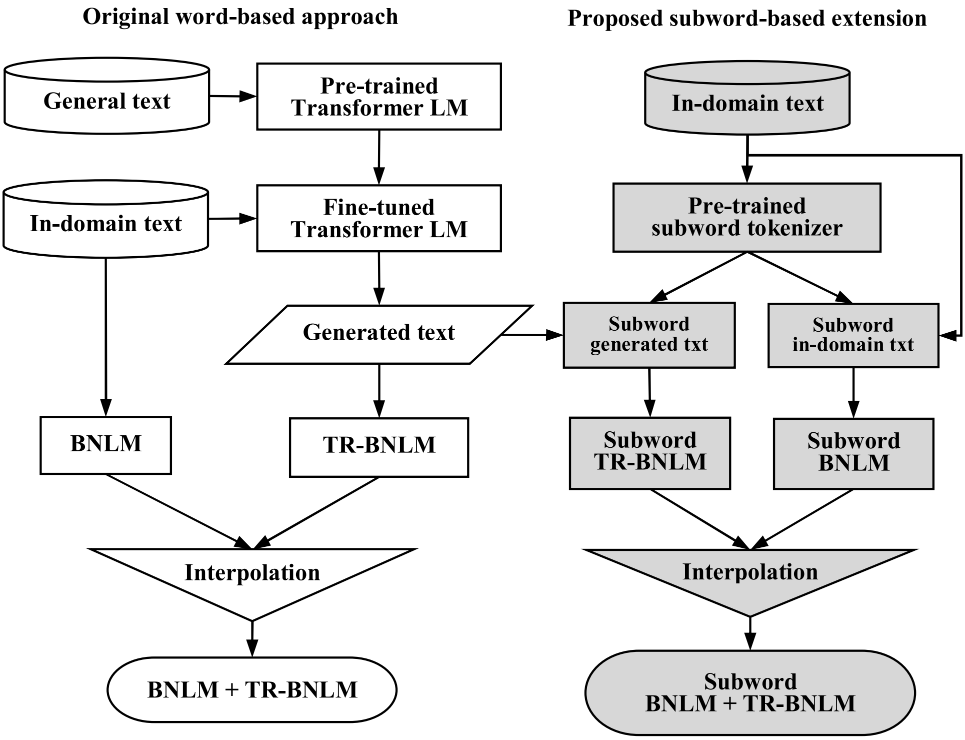

Figure 4. Neural text generation based data augmentation of language models with the proposed modification (gray boxes)

data augmentation process starts with training the subword tokenizer (Morfessor or BPE) on the in-domain dataset. The word-based generated text corpus and the in-domain training text are then segmented into subword sequences using the pre-trained tokenizer. In order to preserve word boundary information during the ASR decoding process, non-initial subwords were tagged with the '+' sign. Based on the segmented text, we train BNLM models (Subword BNLM and Subword TR-BNLM in Figure 4), which can be interpolated for the best performance again (Subword BNLM + TR-BNLM).

5. EXPERIMENTAL RESULTS

In this section, we utilize the techniques presented in Section 4 to show whether the application of neural language model based data augmentation can turn to reduction in WER.

5.1. Experimental setup

High resolution MFCC vectors were used as input features for an LF-MMI trained Factored Time Delay Neural Network (TDNN-F) acoustic model (Povey et al. 2018, 2011). The matrix size (hidden-layer dimension) was 768 and the linear bottleneck dimension was 80 resulting in a total number of 6M parameters in the 12 hidden layers. Phoneme-based acoustic and language model resources were compiled into WFSTs and decoded with the VoXerver (Tarján et al. 2011) ASR decoder. The typical latency of the online decoding setup was measured to be around 250 ms. After experimenting with different n-gram orders on the development set, we found 4-g models the optimal choice both for word and subword BNLMs.

In Hungarian there is a close correspondence between the surface (orthographic) form of words and their phonetic transcription, therefore the pronunciation of most words can be derived by using simple grapheme-phoneme mapping rules (Mihajlik et al. 2002). However, there are always exceptions e.g. foreign words, named entities which are handled with an exception dictionary collected from manual transcripts. Phonetic transcription of subword units are derived by using the same grapheme-phoneme mapping rules as in case of words. In order to handle irregularities, a dictionary of exceptionally pronounced subwords is created by tokenizing the word-based exceptions. The entries of the subword-based exception dictionary are applied as alternative pronunciations in addition to the ones that were generated with the grapheme-phoneme rules.

5.2. Data augmentation with RNNLM

We performed single-pass decoding with the BNLM and RNN-BNLM models and calculated WER of each output (see Table 2). In order to ensure the fair comparison among the modeling approaches, we pruned each RNN-BNLM so that they had similar runtime memory

Table 2. WER of the online ASR system using data augmentation with RNNLM

Token type	Model	Memory usage [GB]	PPL [–]	WER [%]	WERR over Word/Sub. BNLM [%]	
Word	BNLM	1.3	101.1	21.9		
	RNN-BNLM 100M	0.9	102.2	22.5	−2.6*	
	BNLM + RNN-BNLM 100M	1.5	95.7	21.3	2.7*	
Subword	BNLM	1.0	83.7	21.1	3.4*	
	RNN-BNLM 100M	1.1	84.8	21.1	3.7*	0.3
	RNN-BNLM 1B	0.9	78.6	20.5	6.4*	3.2*
	BNLM + RNN-BNLM 100M	1.1	77.1	20.4	6.8*	3.5*
	BNLM + RNN-BNLM 1B	1.1	75.9	20.2	7.7*	4.5*
		3.9	72.8	19.9	8.8*	5.6*

* sign indicates significant difference compared to Word or Subword-based BNLM models and was tested with Wilcoxon signed-rank test ($P < 0.05$).

footprint as the baseline BNLM models (≈1 GB). ASR results of word-based language models show similar trends as perplexity results. The BNLM approximation of word-based RNNLM (Word RNN-BNLM 100M) has a slightly higher WER than the baseline BNLM; however, the interpolated model (Word BNLM + RNN-BNLM 100M) outperforms both. The relative WER improvement of the interpolated model compared to the baseline BNLM is around 3%.

Replacing words with Morfessor derived subwords in the baseline BNLM yields 3% relative WER reduction (Word BNLM vs. Subword BNLM), which is in accordance with our former findings in Tarján et al. (2013). The LM trained on the 100-million-subword generated corpus (Subword RNN-BNLM 100M) has the same WER as the subword-based BNLM (21.1% WER). Using a ten times larger corpus to train the approximated model, however, seems to change the trend. Subword-based RNN-BNLM 1B model is the first approximated model that outperforms a baseline BNLM by itself without interpolation. This observation underlines the importance of the size of the generated text. The difference between 100M and 1B models are also reflected in their interpolated counterparts. Subword BNLM + RNN-BNLM 1B model can reduce WER of subword-based BNLM by ≈5% or even ≈6% if runtime memory consumption is not a restricting factor.

All in all, by augmenting the in-domain training text with a subword-based RNNLM, we managed to reserve real-time operation of the system and reduce the word error rate by 8–9% relative.

5.3. Word-based data augmentation with Transformer LM

5.3.1. Comparing language modeling approaches. In our first experiment, we use the augmentation text generated with the GPT-2 Transformer LM (see Section 4.2) in its original form without subword segmentation. Our goal is to compare the modeling capabilities of language models on in-vocabulary words, hence we limited the vocabulary of all models to the 100k words occurring in the in-domain training text. Previously (see Section 5.2), we augmented the same dataset with a corpus generated by the 2-layer LSTM RNNLM. The results of word-based data augmentation with the RNNLM are placed here to serve as an advanced baseline (RNN-BNLM). All models were pruned to 1 GB runtime memory usage. The results are summarized in Table 3.

Without LM interpolation, neither the RNN-BNLM nor the TR-noPT-BNLM (Transformer without pre-training) models can outperform the baseline BNLM. Only the pre-trained TR-BNLM can reduce the word error rate by around 2% relative. In contrast, with LM interpolation, all augmentation methods reduce significantly the WER of the baseline model. Using a recurrent model (BNLM + RNN-BNLM 100M) or the non-pretrained Transformer (BNLM + TR-noPT-BNLM 1B) for data augmentation result in similar Word Error Rate Reduction (WERR), with the Transformer model being slightly better (2.7% vs. 3.7% WERR). The pre-trained Transformer (BNLM + TR-BNLM 1B), however, stands out among all other approaches, since it reduces the error rate by relative 6%. We also tested whether word-based augmentation models can support each other and found that by applying RNN-BNLM and TR-BNLM simultaneously an additional 1% of WERR can be obtained.

Table 3. WER of the baseline and neural augmented language models using word-based modeling and 100k vocabulary

Model	PPL [–]	OOV rate [%]	WER [%]	WERR [%]
BNLM	101.1	1.8	21.9	-
RNN-BNLM 100M	102.2	1.8	22.5	−2.6*
TR-noPT-BNLM 1B	108.1	1.8	23.1	−5.3*
TR-BNLM 1B	107.3	1.8	21.5	1.7*
BNLM + RNN-BNLM 100M	95.7	1.8	21.3	2.7*
BNLM + TR-noPT-BNLM 1B	94.3	1.8	21.1	3.7*
BNLM + TR-BNLM 1B	90.2	1.8	20.6	5.9*
BNLM + TR-BNLM 1B +RNN-BNLM 100M	87.8	1.8	20.4	6.8*

* sign indicates significant difference compared to BNLM and was tested with Wilcoxon signed-rank test ($P < 0.05$).

5.3.2. Extended word-based augmentation. In the previous section, we limited the vocabulary size of language models to 100k and pruned them to a maximum memory footprint of 1 GB for comparability reasons. In the following, we examine the performance of word-based augmented models without these limitations (See Figure 5).

As it can be seen in a morphologically rich language like Hungarian, the 100k vocabulary size is a strict limitation. By increasing the vocabulary size to 300k, we can reduce the WER by a relative 2% (from 20.6% to 20.2%) and by raising it to 1M by a relative 3% (from 20.6% to 20.0%). If we reduce LM pruning and let the memory footprint to increase from 1 to 4 GB, the WERR can go up to 4.5% (WER from 20.6% to 19.7%), but for such a great improvement we need an extremely large vocabulary with 3 million words. We can see that in a morphologically rich language, exploiting full advantages of neural text generation based data augmentation sacrifices footprint, as large vocabulary and high memory consumption are produced, which severely limits the practical applicability of the approach.

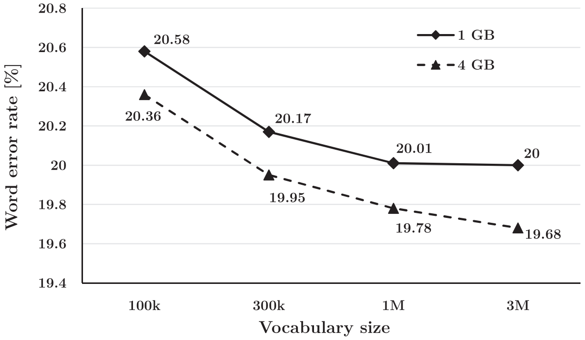

Figure 5. WER of word-based BNLM + TR-BNLM 1B with extended vocabulary and memory footprint

5.4. Subword-based data augmentation with Transformer LM

In order to lower the resource requirements of the augmented language model and utilize the generated text more efficiently, we apply subword LMs (see Section 4.2). Retokenization of the Transformer generated corpus was performed with two different character-level tokenizers: Morfessor and BPE algorithms. In order to make their comparison fair, both tokenizers are trained to use 30k subword units (just like the byte-level tokenizer of the Transformer model). The ASR system models word boundaries with tagged subword units (see Section 2.1), hence the subword language models apply altogether 40k vocabulary items. While in the word-based case the OOV ratio is around 0.6% even with an extremely large 3-million-word vocabulary, the subword-based augmented language models (Subword BNLM + TR-BNLM 1B) can fully cover the test set (0% OOV ratio) with only this 40k subword units.

As shown in Table 4, subword-based data augmentation with a Transformer LM is more effective than with a RNNLM. Morfessor tokenizer slightly outperforms BPE algorithm, however the difference is not statistically significant. Subword modeling can reduce the WER of the 100k word-based model by up to 5% (from 20.6% to 19.6%). The subword BNLM + TR-BNLM, moreover, outperforms the 3-million-word vocabulary word-based model by reducing WER by 2% relative (from 20.0% to 19.6%). Both former improvements were found statistically significant ($P < 0.05$). The WER of the subword-based model with 40k vocabulary and 1 GB memory consumption is roughly the same as the WER of the word-based model with 3M vocabulary items and 4 GB memory usage (19.6% vs. 19.7% WER). Thus, we can state that neural text generation based data augmentation with subword tokenization can

Table 4. WER and PPL of word and subword-based augmentation with normal (1 GB) and extended memory footprint (4 GB)

Model	Vocab size	PPL [–]		OOV rate [%]	WER [%]	
		1 GB	4 GB		1 GB	4 GB
Word BNLM + TR-BNLM 1B	100k	90.2	84.9	1.4	20.6	20.4
	300k	94.2	88.7	1.0	20.2	20.0
	1M	97.5	91.8	0.8	20.0	19.8
	3M	100.8	94.2	0.6	20.0	19.7
Subword – Morfessor BNLM + RNN-BNLM 1B	40k	75.9	72.8	0.0	20.2	19.9
Subword – BPE BNLM + TR-BNLM 1B	40k	69.6	66.3	0.0	19.7	19.4
Subword – Morfessor BNLM + TR-BNLM 1B	40k	69.1	64.7	0.0	19.6	19.3
Subword – Morfessor BNLM + TR-BNLM 1B + RNN-BNLM 1B	40k	67.2	63.5	0.0	19.4	19.1

be significantly more efficient than word-based augmentation for a morphologically rich ASR task.

Just like in the case of word-based models (see Section 5.3.1), using both RNNLM and Transformer models simultaneously, we were able to achieve an additional average relative WER reduction of 1%.

5.5. OOV recognition analysis

The Transformer LM applied in our experiments use subword tokenization, so it can create new word forms when generating text for data augmentation. Hence not only subword-based language models, but word-based models augmented with the Transformer LM become to some extent capable of recognizing out-of-vocabulary words. In this section, we compare this OOV recognition capability of the augmentation approaches. We consider OOV words to be those words that did not occur in the original in-domain training text (see Section 2). We evaluated the ASR outputs of word and subword-based augmentation approaches using information retrieval metrics (Precision, Recall, F_1) (Fawcett 2006).

The results are summarized in Figure 6. The baseline BNLM and the word-based BNLM + TR-BNLM vocab 100k models are not shown in the figure, since they are (obviously) not capable of recognizing OOV words. As it can be seen in Figure 6, all models recognize OOVs with high precision, so it is not typical that OOV words get inserted or replace other words in the ASR transcript. What shows a significant difference between the systems examined is the value of the recall. As the vocabulary size of word-based models increases, so does the recall of OOV words. The 3-million-word vocabulary word-based augmented LM is capable of recognizing almost 22% of OOVs. Subword-based approaches, however, outperform word-based augmentation, while using significantly less resources. Subword BNLM + RNN-BNLM is only slightly better than the best word-based model. However, the subword-based BNLM + TR-

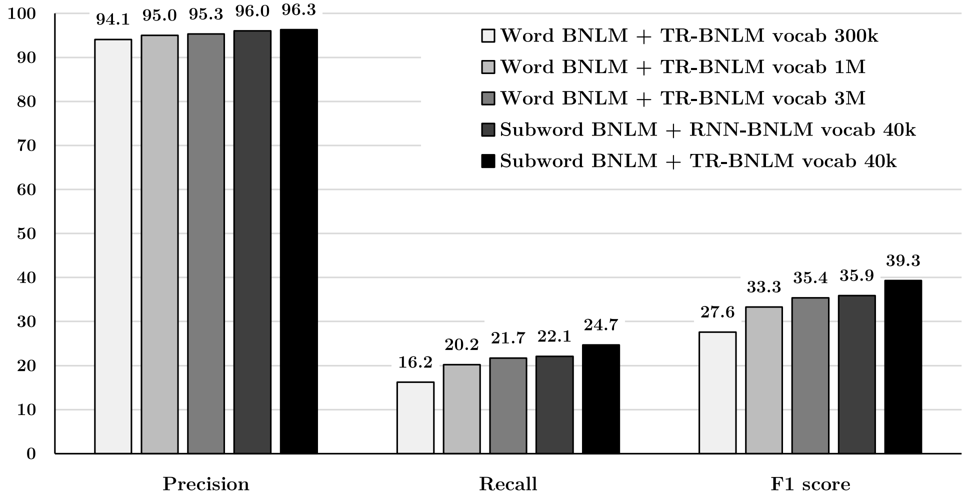

Figure 6. Precision, recall and F_1 of OOV word recognition with various augmented language models

BNLM system can capture every 4th OOV word (≈25% recall) with only 40k subwords in its vocabulary.

6. CONCLUSIONS

In this paper our aim was to improve our Hungarian conversational telephone speech recognition system by handling morphological richness of the language and transferring information from a recurrent and Transformer neural language model to the back-off n-gram model used in single-pass decoding. We compared various types of word-based and subword-based data augmentation techniques and found that by generating a 1-billion-subword corpus with a RNNLM, we were able to achieve 8% relative WER reduction and preserve real-time operation of our ASR system.

We also introduced an approach called *subword-based neural text augmentation* that is the extension of the Transformer based language model augmentation method presented in Wang et al. (2019) for morphologically rich languages. With this new approach we managed to further improve the WER of our online ASR system with 3% relative. Our solution also outperforms the original, word-based data augmentation technique in terms of WER and OOV recognition capability while keeping the vocabulary size and memory requirements of the system quite low.

In the future, we would like to extend our work to other languages and ASR tasks to confirm multilingual portability and task independence of the proposed techniques.

ACKNOWLEDGMENTS

The research was supported by the NKFIH FK-124413 and NKFIH K_20-135038 projects and the Ministry of Innovation and Technology NRDI Office within the framework of the Info-communication and Information Technology National Laboratory Program.

REFERENCES

Adel, Heike, Katrin Kirchhoff, Ngoc Thang Vu, Dominic Telaar and Tanja Schultz. 2014. Comparing approaches to convert recurrent neural networks into backoff language models for efficient decoding. Interspeech 2014. 651–655.

Arisoy, Ebru, Stanley F. Chen, Bhuvana Ramabhadran and Abhinav Sethy. 2014. Converting neural network language models into back-off language models for efficient decoding in automatic speech recognition. IEEE Transactions on Audio, Speech and Language Processing 22(1). 184–192.

Chelba, Ciprian, Mohammad Norouzi and Samy Bengio. 2017. N-gram language modeling using recurrent neural network estimation. arXiv:1703.10724.

Chen, Stanley F. and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. Computer Speech & Language 13(4). 359–393.

Creutz, Mathias and Krista Lagus. 2002. Unsupervised discovery of morphemes. Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning, Vol. 6. 21–30.

Dar, Yehuda, Vidya Muthukumar and Richard G. Baraniuk. 2021. A farewell to the bias-variance tradeoff? An overview of the theory of overparameterized machine learning. arXiv:2109.02355.

Deoras, Anoop, Tomas Mikolov, Stefan Kombrink, Martin Karafiat and Sanjeev Khudanpur. 2011. Variational approximation of long-span language models for LVCSR. 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 5532–5535.

Döbrössy, Bálint, Márton Makrai, Balázs Tarján and György Szaszák. 2019. Investigating sub-word embedding strategies for the morphologically rich and free phrase-order Hungarian. In I. Augenstein, S. Gella, S. Ruder, K. Kann, B. Can, J. Welbl, A. Conneau, X. Ren and M. Rei (eds.) Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019). Stroudsburg, PA: Association for Computational Linguistics. 187–193.

Elman, Jeffrey L. 1990. Finding structure in time. Cognitive Science 14(2). 179–211.

Enarvi, Seppo, Peter Smit, Sami Virpioja and Mikko Kurimo. 2017. Automatic speech recognition with very large conversational Finnish and Estonian vocabularies. IEEE/ACM Transactions on Audio Speech and Language Processing 25(11). 2085–2097.

Fawcett, Tom. 2006. An introduction to ROC analysis. Pattern Recognition Letters 27(8). 861–874.

Hochreiter, Sepp and Jürgen Schmidhuber. 1997. Long short-term memory. Neural Computation 9(8). 1735–1780.

Irie, Kazuki, Albert Zeyer, Ralf Schlüter and Hermann Ney. 2019. Language modeling with deep Transformers. Interspeech 2019. 3905–3909.

Kingma, Diederik P. and Jimmy Ba. 2014. Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations, ICLR 2015 – Conference Track Proceedings. 1–15.

Kurimo, Mikko, Antti Puurula, Ebru Arisoy, Vesa Siivola, Teemu Hirsimäki, Janne Pylkkönen, Tanel Alumäe and Murat Saraclar. 2007. Unlimited vocabulary speech recognition for agglutinative languages. In R.C. Moore, J. Bilmes, J. Chu-Carroll and M. Sanderson (eds.) HTL-NAACL 2006. Stroudsburg, PA: Association for Computational Linguistics. 487–494.

Mihajlik, Péter, Tibor Révész and Péter Tatai. 2002. Phonetic transcription in automatic speech recognition. Acta Linguistica Hungarica 49(3–4). 407–425.

Mihajlik, Péter, Zoltán Tüske, Balázs Tarján, Bottyán Németh and Tibor Fegyó. 2010. Improved recognition of spontaneous Hungarian speech—Morphological and acoustic modeling techniques for a less resourced task. IEEE Transactions on Audio, Speech, and Language Processing 18(6). 1588–1600.

Mikolov, Tomas, Martin Karafiat, Lukas Burget, Jan Cernocky and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. Interspeech 2010. 1045–1048.

Povey, Daniel, Gaofeng Cheng, Yiming Wang, Ke Li, Hainan Xu, Mahsa Yarmohammadi and Sanjeev Khudanpur. 2018. Semi-orthogonal low-rank matrix factorization for deep neural networks. Interspeech 2018. 3743–3747.

Povey, Daniel, 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. 2011. The Kaldi speech recognition toolkit. IEEE 2011 Workshop on Automatic Speech Recognition and Understanding.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. OpenAI Blog 1(8). 9.

Sennrich, Rico, Barry Haddow and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Vol. 1: Long Papers. 1715–1725.

Singh, Mittul, Sami Virpioja, Peter Smit and Mikko Kurimo. 2019. Subword RNNLM approximations for out-of-vocabulary keyword search. Interspeech 2019. 4235–4239.

Smit, Peter, Sami Virpioja and Mikko Kurimo. 2017. Improved subword modeling for WFST-based speech recognition. Interspeech 2017. 2551–2555.

Stolcke, Andreas. 2002. SRILM – An extensible language modeling toolkit. Proceedings International Conference on Spoken Language Processing. 901–904.

Sundermeyer, Martin, Ralf Schlueter and Hermann Ney. 2012. LSTM neural networks for language modeling. Interspeech 2012. 194–197.

Suzuki, Masayuki, Nobuyasu Itoh, Tohru Nagano, Gakuto Kurata and Samuel Thomas. 2019. Improvements to N-gram language model using text generated from neural language model. ICASSP 2019 – 2019 IEEE International Conference on Acoustics, Speech and Signal Processing. 7245–7249.

Tarján, Balázs, Péter Mihajlik, András Balog and Tibor Fegyó. 2011. Evaluation of lexical models for Hungarian Broadcast speech transcription and spoken term detection. 2nd IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2011).

Tarján, Balázs, Gellért Sárosi, Tibor Fegyó and Péter Mihajlik. 2013. Improved recognition of Hungarian call center conversations. 7th Conference on Speech Technology and Human – Computer Dialogue (SPED 2013). 65–70.

Tarján, Balázs, György Szaszák, Tibor Fegyó and Péter Mihajlik. 2019. Investigation on N-gram approximated RNNLMs for recognition of morphologically rich speech. In C. Martín-Vide, M. Purver and S. Pollak (eds.) Statistical Language and Speech Processing: 7th International Conference, SLSP 2019, Ljubljana, Slovenia, October 14–16, 2019, Proceedings. Cham: Springer. 223–234.

Tarján, Balázs, György Szaszák, Tibor Fegyó and Péter Mihajlik. 2020a. Improving real-time recognition of morphologically rich speech with transformer language model. 2020 11th IEEE International Conference on Cognitive Infocommunications (CogInfoCom). 491–496.

Tarján, Balázs, György Szaszák, Tibor Fegyó and Péter Mihajlik. 2020b. On the effectiveness of neural text generation based data augmentation for recognition of morphologically rich speech. In P. Sojka, I. Kopeček, K. Pala and A. Horák (eds.) Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 12284. Cham: Springer. 437–445.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser and Illia Polosukhin. 2017. Attention is all you need. Advances in Neural Information Processing Systems, Vol. 30. 5999–6009.

Virpioja, Sami, Peter Smit, Stig-Arne Grönroos and Mikko Kurimo. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline (Technical report 25/2013). Espoo: Aalto University.

Wang, Yiren, Hongzhao Huang, Zhe Liu, Yutong Pang, Yongqiang Wang, ChengXiang Zhai and Fuchun Peng. 2019. Improving N-gram language models with pre-trained deep transformer. arXiv:1911.10235.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R’emi Louf, Morgan Funtowicz and Jamie Brew. 2019. HuggingFace’s Transformers: State-of-the-art natural language processing. arXiv:1910.03771.

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. Advances in Neural Information Processing Systems, Vol 32. 5753–5763.

Zaremba, Wojciech, Ilya Sutskever and Oriol Vinyals. 2014. Recurrent neural network regularization. arXiv:1409.2329.