INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Finnish ASR with Deep Transformer Models

Abhilash Jain, Aku Rouhe, Stig-Arne Grönroos, Mikko Kurimo

Department of Signal Processing and Acoustics, Aalto University
firstname.lastname@aalto.fi

Abstract

Recently, BERT and Transformer-XL based architectures have achieved strong results in a range of NLP applications. In this paper, we explore Transformer architectures—BERT and Transformer-XL—as a language model for a Finnish ASR task with different rescoring schemes.

We achieve strong results in both an intrinsic and an extrinsic task with Transformer-XL. Achieving 29% better perplexity and 3% better WER than our previous best LSTM-based approach. We also introduce a novel three-pass decoding scheme which improves the ASR performance by 8%. To the best of our knowledge, this is also the first work (i) to formulate an alpha smoothing framework to use the non-autoregressive BERT language model for an ASR task, and (ii) to explore sub-word units with Transformer-XL for an agglutinative language like Finnish.

Index Terms: speech recognition, language modeling, Transformers, BERT, Transformer-XL

1. Introduction

Language modeling has the most important applications in Natural Language Processing (NLP) especially in downstreaming tasks like Automatic Speech Recognition (ASR). Recurrent Neural Networks (RNN) especially Long Short Term Memory (LSTM) networks [1] have been the typical architecture to language modeling which do achieve strong results. In spite of these results, their fundamental sequential computation constraint has restricted their use in the modeling of long-term dependencies in sequential data. To address these issues the Transformer architecture was introduced [2]. The Transformer relies completely on an attention mechanism to form global dependencies between input and output. It also offers more parallelization and has achieved the state-of-the-art (SOTA) results in language modeling outperforming LSTM models [2].

In recent years, there has been a lot of development based on basic Transformer models particularly on unsupervised pretraining [3, 4, 5, 6, 7, 8] which have set state-of-the art results on multiple NLP benchmarks. One such model architecture has been the Bidirectional Encoder Representations from Transformers (BERT) [4] model which uses a deep bidirectional Transformer architecture. Another architecture of interest is the Transformer-XL (T-XL) [5], which introduces the concept of recurrence in a self-attention model and a novel relative positional embedding scheme.

The choice of language model (LM) in ASR for the last five years has commonly been LSTM based models [9, 10, 11]. Recently, the adaptation of Transformer based models as a LM [12, 13, 14] has been proven to be very successful on improving on the earlier results. The focus though has been mostly on the English language for which abundant data is present. It is interesting to see the performance of these models for an agglutinative language like Finnish, which is morphologically richer when compared to English. In this work, we explore the implementation of two Transformer-based models—BERT and T-XL—as a LM in Finnish ASR.

Firstly, we conduct text-based experiments to see how they perform in word prediction. We take inspiration from recent works [15, 16] in investigating Deep Transformers. A sub-word based approach for both T-XL and BERT is implemented as Finnish has a very large vocabulary. With smaller units, the modeled sequences are longer, and we expect that the recursive XL architecture can allow us to still model long term effects, but avoid the Transformer's memory issues (grows quadratically with size). To the best of our knowledge, this is the first work to use subword units with T-XL for an agglutinative language like Finnish.

Secondly, we perform the Finnish ASR task on the YLE news dataset. The ASR setup is the same as in [17] which is considered the previous best. The same training data as the previous best LSTM is used to train both BERT and T-XL. We compare the Word Error Rate (WER) of all models using N-Best list rescoring. The above is done to ensure fair comparisons among different model architectures. We experiment with a novel rescoring technique which accounts for BERT's bi-directionality and the sharper predicted word probability distribution. This is one of the first works using a BERT rescoring technique with α smoothing.

We are able to successfully apply these models for ASR. T-XL obtained strong results when compared to the interpolation of LSTM+N-Gram used in [17]. Rescoring with BERT did improve the accuracy of the 1st pass ASR system, but was still behind rescoring with the LSTM model. We also develop a three-pass decoding technique where we rescore a short N-best list generated by the 2^{nd} pass LSTM+N-Gram. This technique also achieved strong results when compared to the LSTM+N-Gram and an interpolation of T-XL and LSTM+N-Gram.

2. Methods

2.1. Language Modeling - Perplexity

The goal of a LM is to assign meaningful probabilities to a sequence of words. Given a set of tokens $\mathbf{X} = (x_1, ..., x_T)$, where T is the length of a sequence, our task is to estimate the joint conditional probability $P(\mathbf{X})$ which is

$$P(\mathbf{X}) = \prod_{i=1}^{T} p\left(x_i | x_1, \ldots, x_{i-1}\right), \tag{1}$$

where $(x_1, \ldots, x_{i-1})$ is the context. An intrinsic evaluation of the performance of a LM is perplexity (PPL) which is defined as the inverse probability of the set of the tokens and taking the T^{th} root were T is the number of tokens

$$\text{PPL}(\mathbf{X}) = P(\mathbf{X})^{-1/T}. \tag{2}$$

http://dx.doi.org/10.21437/Interspeech.2020-1784

Calculating the auto-regressive $P(\mathbf{X})$ for the T-XL is quite straight-forward as the model is unidirectional. Due to BERT's non-auto-regressive nature, the joint probability does not factorize the same way.

BERT's bi-directional context poses a challenge for us to calculate an auto-regressive joint probability. A simple workaround could be that we mask all the tokens $\mathbf{x}_{>i}$ and calculate the conditional factors as we do for an unidirectional model. By doing so though, we lose the advantage of bi-directional context the BERT model enables. We use an approximation of the joint probability as,

$$P(\mathbf{X}) \approx \prod_{i=1}^{T} p\left(x_i | x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_T\right). \quad (3)$$

Eq.3 is defined as a pseudo-perplexity (pseudo-PPL) score. This pseudo-PPL is used in our language modeling experiments with BERT. This type of approximations has been previously explored with Bi-directional RNN LM's [18] but not for deep Transformer models. The one drawback [18] with this approach is higher probabilities assigned to each word and one way to address this issue is to use a tunable parameter α to smooth the probability distribution. Then the word probability is,

$$p\left(x_i | x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_T\right) = \frac{\exp\left(\alpha y_i\right)}{\sum_j^T \exp\left(\alpha y_j\right)} \quad (4)$$

where y_i is the activation before the softmax function in the output layer. We set α to 0.6 in all our tasks after optimizing it over the development set. We apply this α smoothing technique with our BERT models during inference. This is advantageous as this technique can be applied with already existing pre-trained BERT models.

2.2. LM training with BERT and Transformer-XL

The original BERT has two training objectives: Masked language modeling (MLM), in which you mask input tokens randomly and then predict the masked tokens using the left and right context. Next, there is the 'next sentence prediction' task that jointly trains text-pair representations. We will drop this objective as it was intended for downstream tasks like 'Question & Answering' and offers little gains for other tasks [19].

T-XL is a unidirectional deep Transformer architecture, therefore the PPL can be calculated as (Eq 2). The only change is in the input format, where we use sub-word units rather than whole word units.

To remain consistent with experiments performed with the previous best LSTM [17] we use Morfessor 2.0 [20, 21] for the subword tokenization in Finnish for both BERT and T-XL. We also apply the same boundary markers- left+right-marked (+m+) markings which was the best performing sub-word marking in [17]. We use the basic unsupervised Morfessor Baseline algorithm [22] with a corpus weight parameter of 0.001. This parameter choice achieved the best result in [17].

2.3. Rescoring with BERT and Transformer-XL

To fairly compare the performances of our BERT, T-XL and LSTM models, we perform ASR experiments. Lattice rescoring is a difficult task for Bi-directional models when compared to uni-directional models [23] and therefore to simplify our approach we will be rescoring on a N-Best list (50-Best in our case). The N-Best candidates are gathered from the first-pass decoding with the same ASR system used by [17]. The N-Best candidates are then further reranked with BERT and T-XL models using the following score:

$$\text{score} = \lambda \cdot \text{score}_{\text{LM}} + \text{score}_{\text{AM}} \quad (5)$$

where score_{LM} and score_{AM} are the score of each hypothesis from the LM and AM, respectively. λ is a LM weight parameter empirically determined from a development set.

While training BERT with MLM, both the left and right context is available. More context can be advantageous to rescore utterances. In our approach, instances are created of the target utterance with one word replaced by the mask token at a time. For example, if the utterance has 4 tokens, we would create 4 instances as in Table 1.

Table 1: *Masked instances for a Finnish word like 'verkko+ +sivu+ +illa+ +an' which translates to 'on their website'*

Instance No.	Label	Masked Input
1	verkko+	[MASK] +sivu+ +illa+ +an
2	+sivu+	verkko+ [MASK] +illa+ +an
3	+illa+	verkko+ +sivu+ [MASK] +an
4	+an	verkko+ +sivu+ +illa+ [MASK]

Next, the BERT LM computes the log-likelihood using Eq.3 of the original label in the masked position. For the α smoothing BERT computation is done using the Eq.4. All the log-likelihoods of each instance is summed up and this total is defined as the score of each sentence. Even though this is not the same as the sentence probability generated by a uni-directional model, this can be used to compute score_{LM} to rescore N-best lists. Next, Eq.5 is used to rerank the N-Best list. This task is parallelized as each instance calculation is independent of each other. Also this rescoring task does not require further training and thus can be directly used with the pre-trained model. Similar approach has been used for a English BERT task [24] except they don't smooth the word probabilities as Eq.4 and they further train the model with task-specific data.

For T-XL, due to its uni-directionality the sentence probabilities are calculated using Eq.1, Next Eq.2 is used to calculate score_{LM}. Eq.5 is used to rerank the N-Best candidates. interpolation of two LM's - LM1 and LM2 is done by modifying the Eq.5 as,

$$\text{score} = \lambda 1 \cdot \text{score}_{\text{LM1}} + \lambda 2 \cdot \text{score}_{\text{LM2}} + \text{score}_{\text{AM}} \quad (6)$$

where $\lambda 1$ and $\lambda 2$ are optimized for each LM.

T-XL model is used to develop a three-pass decoding scheme. This is a pipeline strategy, where the first pass is the same as [17]. In the second pass, LSTM+N-Gram is used to rescore lattices from the first-pass to generate a shorter N-Best list. The rescoring of lattices is the same as [17]. Lastly in the third pass, T-XL is used to rescore the short N-Best list using Eq.5. The second pass prunes a majority of the less likely candidates and the heavy LM does not rescore all the possible N-best candidates in the third pass, potentially saving time and resources. Fig. 1 represents the three-pass decoding scheme.

3. Data

We use 1500 hours of speech containing manually transcribed sessions from the Finnish parliament and read speech from large number of speakers was utilized to train the Acoustic Model

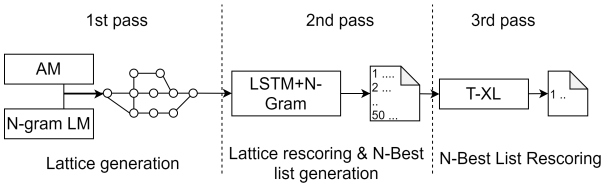

Figure 1: *Schematic diagram of three-pass decoding scheme*

(AM) as described in [17]. The Finnish text data used for all the language modeling experiments is provided by Kielipankki - the Language Bank of Finland [25]. The dataset consists mainly of newspapers and books of around 144 million word tokens and 4.2 million unique types of tokens. The data is preprocessed to remove any punctuation marks, special characters. We also convert digits to numerals. The data is randomly divided into a training and a validation dataset. The validation set contains 10K sentences. In the training data, the average token length per sentence is 21 and the maximum length is 300 per sentence. We have a total of 234 million tokens in 12.8 million sentences with a vocabulary of 34K subword tokens. The input is one sentence per line and we shuffle the sentences at each epoch. For ASR evaluation, The development and test set dataset consists of 2850 and 3006 utterances of Finnish transcribed broadcast news obtained from the Finnish national broadcaster YLE. This test set has been previously used to evaluate Finnish ASR systems in [26, 27, 17]

4. Language Modeling Experiments

All BERT and T-XL models in the experiments are trained for a maximum of 1.2 million steps and an early stopping technique is applied if the training loss keeps increasing. Adam is used as an optimizer for both BERT and T-XL experiment sets. All the experiments are trained on a single NVIDIA Tesla V100 32 GB graphic card unlike the multi-gpu setup used by the English models [4, 5]. Only the models fitting on a single -gpu were selected to keep resource demands low. The previous best was an interpolated LSTM+N-Gram. The LSTM model is taken from [17], it is a 'deep' architecture which starts from a projection and a LSTM layer that has four pairs of dropout and highway layers. The large N-Gram is also taken from [17] and contains 50-80 million n-gram contexts. Hence-forth, we call this interpolation as **LSTM+N-Gram**.

4.1. BERT

BERT models are very resource intensive to train, and therefore our hyperparameter search space is influenced by the original BERT [4]. We also tried to increase the depth rather than width as suggested in [15]. All the BERT models are pre-trained from scratch and the pseudo-perplexities are calculated on the development set. Different configurations using the development set are tried out, finally settling on the configurations in Table 2.

Even though the BERT model pseudo-PPL cannot be directly compared with the T-XL PPL, the pseudo-PPL can still be used to intrinsically evaluate between different BERT models. The 20 layer BERT model from Table 2 performs the best among all the BERT models after α smoothing.

Table 2: *Pseudo-perplexities calculated on the test set for different configurations of the BERT model with layers (L), feed-forward layer size (FF), hidden Transformer size (H) and attention heads (A)*

L	FF	H	A	Pseudo PPL W/o α	Pseudo PPL With α
10	1024	360	8	62.11	952.2
12	3072	768	12	18.86	49.29
4	3584	896	16	20.35	45.1
6				19.74	44.31
8				11.56	28.51
10				**11.48**	28.33
20				11.84	**28.27**

Table 3: *Perplexities calculated on the test set with different model configurations with layers (L), feed-forward layer size (FF), hidden Transformer size (H)*

LM	L	FF	H	seg-mem	PPL
LSTM+N-Gram					93.2
Transformer	18	4096	1024		78.7
T-XL	3	2048	512	150-150	89.6
	4	2048	512	150-150	82.3
	4	4096	1024	32-32	75.1
	8	1024	256	32-32	85.4
	16				75.3
	32				68.7
	64				67.1
	72				**66.3**

4.2. Transformer-XL

Because T-XL utilizes previous context to calculate its attention as explained in [5] the selection of contexts is an important hyperparameter. They are seg-length(seg) and mem-length(mem) as explained in [5] As training T-XL is very resource and time intensive, we focus on comparing a wider context but a shallower model against a narrower context but a deeper model.

The same cosine annealing learning rate scheduler and relative positional embeddings as [5] are used in all of the T-XL experiments. A baseline Transformer model is trained with the same self-attention as BERT and no previous context, the hyperparameters resemble the one's in [5]. All T-XL models are trained from scratch. The perplexities are calculated on the development set and the results are in Table 3.

From Table 3 the T-XL with 72 layers achieved a PPL score of 66.3 achieving a 29% better score than LSTM+N-Gram.

5. ASR Experiments

5.1. Two-pass decoding

First a conventional rescoring scheme is a applied to use the LMs trained in the previous section to rescore a 50-best list in the YLE news dataset. In preliminary experiments we found 50 to be a suitable compromise between speed and accuracy for all LMs. The AM is a TDNN-BLSTM trained with lattice-free MMI and the small first-pass N-gram LM contains approximately 5M N-grams and both the models are from [17]. Table 4

Table 4: *WERs (%) on different LMs obtained by 50-best rescoring (except the 5M N-gram LM)*

LM	WER	
	dev	test
5M N-gram	17.56	25.41
LSTM+N-Gram	14.19	20.57
Baseline Transformer	13.93	20.23
BERT (without α smoothing)		
L-10 FF-3584 H-896 A-16	16.35	25.62
L-20 FF-3584 H-896 A-16	16.25	25.72
BERT (with α smoothing)		
L-10 FF-3584 H-896 A-16	15.39	25.21
L-20 FF-3584 H-896 A-16	15.34	25.26
T-XL		
L-4 FF-4096 H-1024	14.63	21.23
L-8 FF-1024 H-256	14.11	20.54
L-16 FF-1024 H-256	13.81	20.24
L-32 FF-1024 H-256	13.78	**20.04**
L-48 FF-1024 H-256	13.74	20.16
L-64 FF-1024 H-256	13.72	**20.04**
L-72 FF-1024 H-256	**13.67**	20.12

shows that the best T-XL outperforms the LSTM+N-Gram by 5% on the dev set and 3% on the test set. The BERT models do outperform the 5M N-gram but still cannot get a better WER than LSTM+N-Gram. There is an incremental gain in WER by applying the α smoothing. The pseudo-PPL with α and WER results appear to correlate. Therefore, we can potentially use the pseudo-PPL with α to faster optimize the hyperparameters without using the model on ASR.

5.2. Three–pass decoding scheme

A three-pass decoding scheme was tested for the best performing T-XL's from the Sec. 5.1. On the first pass of the ASR, A TDNN-BLSTM AM and a 5M N-gram LM same as Sec.5.1 is used. In the second pass, LSTM+N-gram rescores first pass lattices generating a new 50-best list. In the third pass the 50-best list is rescored using T-XL. We also perform interpolating experiments by using the LM scores from LSTM+N-gram and T-XL after each of them have rescored a 1000-best list generated from the first pass. In preliminary experiments we found that 1000 is a good compromise between speed and accuracy. With T-XL as LM1 and LSTM+N-gram as LM2 in Eq.6, optimal λ_1=2λ_2. This signifies that T-XL is twice more impactful than LSTM+N-gram in the interpolation results. From Table 5, our best three-pass decoding scheme outperforms the Interpolated LSTM+N-gram+T-XL by 4%.

6. Discussion

BERT-based architectures [28, 29] have set very good benchmarks in a variety of tasks like part of speech tagging, named entity recognition, Question & Answering. However, they have not attracted as much attention in ASR, where their results have been much behind the best [24]. This may be due to the fact that BERT has a broader objective like sentence encoding due to the masked LM training and ASR requires the LM to have a auto-regressive training component. Nonetheless, this could be leveraged as during rescoring we do have the entire utterance at hand. Thus, a more efficient rescoring scheme could be developed in the future to take advantage of both T-XL and BERT.

Table 5: *WERs (%) of the following decoding models:*
[1]*rescoring a 1000-best,*
[2]*interpolation after each rescoring 1000-best,* $\lambda_{T\text{-}XL} = 2\lambda_{LSTM}$
[3]*three-pass decoding with LSTM+N-gram rescoring a 1000-best list followed by T-XL rescoring a 50-best, and*
[4]*three-pass decoding with LSTM+N-Gram rescoring a lattice followed by T-XL rescoring a 50-best list*

LM	WER	
	dev	test
[1]**LSTM+N-Gram**	13.68	19.3
[1]**T-XL**		
L-32 FF-1024 H-256	13.10	18.65
L-64 FF-1024 H-256	13.00	18.50
L-72 FF-1024 H-256	12.95	18.63
[2]**T-XL and LSTM+N-Gram (Interpolated)**		
L-32 FF-1024 H-256	12.76	18.54
L-64 FF-1024 H-256	12.78	18.38
L-72 FF-1024 H-256	12.77	18.40
[3]**1000-best LSTM+N-Gram + 50-best T-XL**		
L-32 FF-1024 H-256	13.07	18.51
L-64 FF-1024 H-256	13.02	18.42
L-72 FF-1024 H-256	12.85	18.56
[4]**Lattice LSTM+N-Gram + 50-best T-XL**		
L-32 FF-1024 H-256	12.76	18.05
L-64 FF-1024 H-256	12.68	17.72
L-72 FF-1024 H-256	**12.59**	**17.71**

In our experiments, T-XL performs better when compared to the LSTM+N-Gram both in the intrinsic and extrinsic tasks. T-XL's encapsulation of the recurrence mechanism is advantageous for long sequences of subword units. LSTM+N-Gram and T-XL are also used in the three-pass decoding scheme outperforming both the individual models and the interpolated models on the N-Best lists. In comparison to the interpolated models, the three-pass decoding scheme runs the heavy LM rescoring on a pruned set of hypothesis saving time and space.

7. Conclusion

We apply BERT and T-XL LMs for speech recognition. We show that Transformer-XL outperforms LSTM+N-Gram on perplexity by 29% and ASR by 3% with the same data. We propose a three-pass decoding scheme which successfully approximates the interpolation of LSTM+N-Gram + T-XL and avoids running the large slow T-XL language model for the full 1000-best hypothesis as in the interpolation. We propose a framework for pre-trained BERT models along with α smoothing which can be used for ASR. We believe it is possible to improve the performance of these models by using more data, applying more regularization techniques and scaling them up.

8. Acknowledgements

This work was supported by the Academy of Finland (grant 329267) and EU's Horizon 2020 research and innovation programme via the project MeMAD (GA 780069). The computational resources were provided by Aalto ScienceIT.

9. References

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*. Curran Associates, Inc., 2017, pp. 5998–6008. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[3] A. Radford, "Improving language understanding by generative pre-training," 2018.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[5] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," in *ACL*, 2019.

[6] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 5753–5763. [Online]. Available: http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding.pdf

[7] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 2227–2237.

[8] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 328–339.

[9] K. J. Han, A. Chandrashekaran, J. Kim, and I. Lane, "The CAPIO 2017 Conversational Speech Recognition System," *arXiv e-prints*, p. arXiv:1801.00059, Dec 2017.

[10] G. Saon, T. Sercu, S. Rennie, and H.-K. J. Kuo, "The ibm 2016 english conversational telephone speech recognition system," *Interspeech 2016*, Sep 2016. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2016-1460

[11] G. Kurata, B. Ramabhadran, G. Saon, and A. Sethy, "Language modeling with highway lstm," *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Dec 2017. [Online]. Available: http://dx.doi.org/10.1109/ASRU.2017.8268942

[12] C. Lüscher, E. Beck, K. Irie, M. Kitza, W. Michel, A. Zeyer, R. Schlüter, and H. Ney, "Rwth asr systems for librispeech: Hybrid vs attention," *Interspeech 2019*, Sep 2019. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1780

[13] Y. Wang, A. Mohamed, D. Le, C. Liu, A. Xiao, J. Mahadeokar, H. Huang, A. Tjandra, X. Zhang, F. Zhang, C. Fuegen, G. Zweig, and M. L. Seltzer, "Transformer-based acoustic modeling for hybrid speech recognition," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6874–6878.

[14] I. Medennikov, Y. Khokhlov, A. Romanenko, I. Sorokin, A. Mitrofanov, V. Bataev, A. Andrusenko, T. Prisyach, M. Korenevskaya, O. Petrov, and A. Zatvornitskiy, "The STC ASR System for the VOiCES from a Distance Challenge 2019," in *Proc. Interspeech 2019*, 2019.

[15] K. Irie, A. Zeyer, R. Schlüter, and H. Ney, "Language Modeling with Deep Transformers," in *Proc. Interspeech 2019*, 2019, pp. 3905–3909. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2225

[16] J. Li, V. Lavrukhin, B. Ginsburg, R. Leary, O. Kuchaiev, J. M. Cohen, H. Nguyen, and R. T. Gadde, "Jasper: An End-to-End Convolutional Neural Acoustic Model," in *Proc. Interspeech 2019*, 2019, pp. 71–75. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1819

[17] P. Smit, "Modern subword-based models for automatic speech recognition," ser. Aalto University publication series DOCTORAL DISSERTATIONS; 97/2019. Aalto University; Aalto-yliopisto, 2019, G5 Artikkeliväitöskirja, pp. 62 + app. 136. [Online]. Available: http://urn.fi/URN:ISBN:978-952-60-8566-1

[18] X. Chen, A. Ragni, X. Liu, and M. Gales, "Investigating bidirectional recurrent neural network language models for speech recognition," 08 2017, pp. 269–273.

[19] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: http://arxiv.org/abs/1907.11692

[20] M. Creutz and K. Lagus, "Unsupervised discovery of morphemes," in *ACL-02 Workshop on Morphological and Phonological Learning*, ser. MPL '02, vol. 6. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 21–30. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1118647.1118650

[21] S. Virpioja, P. Smit, S.-A. Grönroos, and M. Kurimo, "Morfessor 2.0: Python implementation and extensions for morfessor baseline," Department of Signal Processing and Acoustics, Aalto University, Helsinki, Finland, Report 25/2013 in Aalto University publication series SCIENCE + TECHNOLOGY, 2013.

[22] M. Creutz and K. Lagus, "Unsupervised models for morpheme segmentation and morphology learning," *ACM Trans. Speech Lang. Process.*, vol. 4, no. 1, Feb. 2007. [Online]. Available: https://doi.org/10.1145/1187415.1187418

[23] X. Liu, Y. Wang, X. Chen, M. J. F. Gales, and P. C. Woodland, "Efficient lattice rescoring using recurrent neural network language models," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4908–4912.

[24] J. Shin, Y. Lee, and K. Jung, "Effective sentence scoring method using bert for speech recognition," in *Proceedings of The Eleventh Asian Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, W. S. Lee and T. Suzuki, Eds., vol. 101. Nagoya, Japan: PMLR, 17–19 Nov 2019, pp. 1081–1093.

[25] CSC - IT Center for Science, "The helsinki korp version of the Finnish text collection, url: http://urn.fi/urn:nbn:fi:lb-2016050207," 1998. [Online]. Available: http://urn.fi/urn:nbn:fi:lb-2016050207

[26] A. Mansikkaniemi, P. Smit, and M. Kurimo, "Automatic construction of the finnish parliament speech corpus," in *Proc. Interspeech 2017*, 2017, pp. 3762–3766. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-1115

[27] M. Varjokallio, S. Virpioja, and M. Kurimo, "First-pass techniques for very large vocabulary speech recognition of morphologically rich languages," in *2018 IEEE Spoken Language Technology Workshop, December 18-21, 2018, Athens, Greece*. United States: IEEE, 2018, pp. 227–234.

[28] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=H1eA7AEtvS

[29] A. Virtanen, J. Kanerva, R. Ilo, J. Luoma, J. Luotolahti, T. Salakoski, F. Ginter, and S. Pyysalo, "Multilingual is not enough: Bert for finnish," *ArXiv*, vol. abs/1912.07076, 2019.