Interspeech 2022
18-22 September 2022, Incheon, Korea

Effect and Analysis of Large-scale Language Model Rescoring on Competitive ASR Systems

Takuma Udagawa[1], *Masayuki Suzuki*[1], *Gakuto Kurata*[1], *Nobuyasu Itoh*[1], *George Saon*[2]

[1]IBM Research - Tokyo, Japan
[2]IBM T. J. Watson Research Center, Yorktown Heights, USA

Takuma.Udagawa@ibm.com, {szuk,gakuto,iton}@jp.ibm.com, gsaon@us.ibm.com

Abstract

Large-scale language models (LLMs) such as GPT-2, BERT and RoBERTa have been successfully applied to ASR N-best rescoring. However, whether or how they can benefit competitive, near state-of-the-art ASR systems remains unexplored. In this study, we incorporate LLM rescoring into one of the most competitive ASR baselines: the Conformer-Transducer model. We demonstrate that consistent improvement is achieved by the LLM's bidirectionality, pretraining, in-domain finetuning and context augmentation. Furthermore, our lexical analysis sheds light on how each of these components may be contributing to the ASR performance.

Index Terms: speech recognition, large-scale language models, N-best rescoring

1. Introduction

Large-scale language models (LLMs) such as GPT-2, BERT and RoBERTa [1–3] have become a prominent component in modern NLP. Based on their self-attention mechanism [4], Transformer-based LLMs are capable of modeling long-range dependencies and interactions within the input, which are essential for higher-level language understanding. Conventionally, LLMs are first *pretrained* on massive unlabelled text data, which allows them to learn the general linguistic knowledge (e.g. morphosyntax and semantics) [5, 6] and world knowledge [7]. Subsequently, these models can be *finetuned* with a smaller amount of labelled data to achieve superior performance on specific (narrower) target tasks and domains [8].

To leverage the power of LLMs into ASR, recent work has proposed to rescore the ASR model's N-best hypotheses based on LLMs [9–15] in place of the traditional *n*-gram or RNN-based LMs [16–18]. In the most simple and foundational approach [9, 10], LLM scores are computed as the *log-likelihood* of each hypothesis for unidirectional models (e.g. GPT-2) and *pseudo-log-likelihood* for bidirectional models (e.g. BERT and RoBERTa). By incorporating such LLM scores, we can expect to improve the linguistic acceptability of the final ASR results, including utterance-level consistency and discourse-level consistency across utterances [19, 20].

However, existing empirical studies are conducted on modest ASR baselines with considerable room left for improvement, and it remains unclear whether or how LLM rescoring works in more competitive ASR systems. Therefore, in this study, we examine the effect of LLM rescoring on a strong Conformer-Transducer model [21], which is a competitive, near state-of-the-art ASR baseline among non-attention-based models [22]. Furthermore, we assess how LLM rescoring improves ASR through a simple lexical analysis, where we decompose the error reduction rates based on word frequency and error type.

Based on our experiments with the widely used Switchboard dataset, we show that bidirectional LLMs (BERT and RoBERTa) consistently improve ASR results, while the unidirectional LLM (GPT-2) fails to do so on our competitive baseline. In addition, LLM's pretraining, in-domain finetuning and context augmentation (with past and future hypotheses) also consistently improves ASR. Finally, our lexical analysis sheds light on how each of these variants may be contributing to the ASR results, often providing complementary benefits.

2. Methods

In this work, we follow the most simple and foundational approach of LLM rescoring proposed in [9, 10]. The main idea is to compute the LLM score as the *log-likelihood* for unidirectional models and *pseudo-log-likelihood* for bidirectional models, as we will explain in Section 2.1 and 2.2.

2.1. Unidirectional LLM Scoring

Unidirectional language models, also known as *causal language models*, are trained to predict the conditional probability of the next word given the prior history of words. To be precise, a unidirectional LLM with parameters Θ is trained to predict the probability $P_{\mathrm{LM}}(\boldsymbol{w}_t|\boldsymbol{W}_{<t};\Theta)$, where $\boldsymbol{W}_{<t} := (\boldsymbol{w}_1, ..., \boldsymbol{w}_{t-1})$ denotes the prior history of words up to $\boldsymbol{w}_t$. Then, a natural choice of the LLM score for the hypothesis $\boldsymbol{W} := (\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_{|\boldsymbol{W}|})$ would be its *log-likelihood*, which can be computed by the chain rule of probability:

$$\mathrm{Score}_{\mathrm{LM}}(\boldsymbol{W}) := \sum_{t=1}^{|\boldsymbol{W}|} \log P_{\mathrm{LM}}(\boldsymbol{w}_t|\boldsymbol{W}_{<t};\Theta) \qquad (1)$$

Note that such LLMs are *unidirectional* in the sense that prediction of $\boldsymbol{w}_t$ only depends on the previous (left) context. One advantage of unidirectional LLM scoring is its computational efficiency, only requiring a single inference pass to compute the log-likelihood in an autoregressive manner.

2.2. Bidirectional LLM Scoring

In contrast, bidirectional language models aim to predict $\boldsymbol{w}_t$ conditioned on both the left and right context. One representative example of bidirectional models are the *masked language models*, which can be trained to estimate the conditional probability $P_{\mathrm{LM}}(\boldsymbol{w}_t|\boldsymbol{W}_{\setminus t};\Theta)$, where $\boldsymbol{W}_{\setminus t} := (\boldsymbol{w}_1, ..., \boldsymbol{w}_{t-1}, \texttt{[MASK]}, \boldsymbol{w}_{t+1}, ..., \boldsymbol{w}_{|\boldsymbol{W}|})$ denote the hypothesis $\boldsymbol{W}$ with the word $\boldsymbol{w}_t$ replaced by a special `[MASK]` token. Unfortunately, based on this likelihood, it is non-trival to estimate the exact log-likelihood of $\boldsymbol{W}$. Instead, prior work proposed to use its *pseudo-log-likelihood* [23] as the LLM score,

10.21437/Interspeech.2022-11123

which is given by the following equation:

$$\text{Score}_{\text{LM}}(\boldsymbol{W}) := \sum_{t=1}^{|\boldsymbol{W}|} \log P_{\text{LM}}(\boldsymbol{w}_t | \boldsymbol{W}_{\setminus t}; \Theta) \quad (2)$$

Note that by leveraging the *bidirectional* (left and right) context, prediction of each $\boldsymbol{w}_t$ becomes more accurate, which can lead to more reliable LLM scoring. However, this comes with the cost of running the model $|\boldsymbol{W}|$ times (with each $\boldsymbol{w}_t$ replaced by the `[MASK]` token), which can be slow or computationally expensive.

Based on the LLM scores from equation (1) or (2), each hypothesis in the N-best can be rescored by the final score, which is a linear interpolation with the ASR model (AM) score:

$$\text{Score}(\boldsymbol{W}) := (1-\lambda) \cdot \text{Score}_{\text{AM}}(\boldsymbol{W}) + \lambda \cdot \text{Score}_{\text{LM}}(\boldsymbol{W}) \quad (3)$$

3. Experiments

3.1. LLM Rescoring

For LLM rescoring, we use the open-source GPT-2, BERT and RoBERTa from the huggingface library [24]. While the pretrained models can be used out-of-the-box based on our method (Section 2), we also experiment with the following configurations to study the effect of LLM *pretraining*, *in-domain finetuning* and *context augmentation*, respectively.

Training from Scratch (+*scratch*): To ablate the general knowledge acquired in the pretraining step, we train each LLM from scratch (i.e. using the same model architecture with randomly initialized parameters) based on the transcribed text of 2000 hours of Switchboard corpus.

In-domain Finetuning (+*finetune*): While the pretraining corpora of LLMs cover diverse domains and topics, they mostly consist of written (non-spoken) language. To examine the additional gain from speech domain adaptation, we use the pretrained model parameters and finetune each LLM based on the same 2000 hours text of Switchboard corpus.

Context Augmentation (+*context*): Without changing the model parameters, we concatenate the 1-best of past and future hypotheses as additional context when computing the LLM scores [13]. For the past hypotheses, the 1-best is obtained (and cached) after LLM rescoring with a fixed λ value of 0.20 (from equation (3)). For the future hypotheses, the 1-best is obtained from the initial ASR model. We use up to 40 tokens from the past 1-best and 20 tokens from the future 1-best, concatenated on the left and right of the current hypothesis.

Based on the results from our preliminary experiments, we insert special tokens at the beginning and ending (e.g. `[CLS]` and `[SEP]` for BERT) and do not indicate utterance boundaries (e.g. with a period [13]) during context augmentation. Focusing on the ceiling performance, we choose the best (and largest) λ from $\{0.0, 0.05, ..., 1.00\}$ for equation (3) which achieve the lowest average WER on our test sets.

3.2. ASR Baseline

To confirm the advantage of LLM rescoring, we trained a strong baseline neural transducer model. We used the widely used 1975 hours of English conversational telephone speech consisting of Switchboard, Fisher, and CallHome as training data [25]. We created 9875 hours of augmented training data by applying speed and tempo perturbation (90% and 110% for both perturbation) to the original 1975 hours [26].

For acoustic features, we used 240-dimensional features consisting of 40-dimensional log-Mel filterbank energies, their delta, and double-delta coefficients with frame stacking and a skipping rate of 2 [27]. Training of the baseline model followed a recipe [28] that leverages various regularization techniques such as SpecAugment [29], sequence noise injection [30], DropConnect [31], and speed and tempo perturbation [26]. The Conformer-T model uses an encoder network of 10 Conformer blocks (512-dimensional feed forward module, 31-kernel convolution block, and 8 64-dimensional attention heads) [21]. The encoder output is projected from a 512-dimensional acoustic embedding to 256 dimensions. A prediction network uses a single unidirectional LSTM layer with 1024 cells followed by a projection from a 1024-dimensional linguistic embedding to 256 dimensions. A joint network uses multiplicative integration and hyperbolic tangent, followed by a projection and softmax layer to represent 42 characters and an extra blank symbol. We used an efficient beam search algorithm, alignment-length synchronous decoding [32] for letter-based decoding, and used Word Error Rate (WER) as the evaluation metric.

Based on this Conformer-T model, we obtain the *100-best hypotheses* searched with a beam size of 100 and a more realistic *16-best hypotheses* searched with beam size 16. We report our baseline and LLM rescored results on the Hub5 2000 Switchboard (SWB) and CallHome (CH) evaluation test sets.

4. Results

We show the LLM rescoring results with 16-best hypotheses in Table 1 and 100-best hypotheses in Table 2. Results for the GPT-2 models are omitted since they did not improve WER (0.1 or more) with any value of λ or model configurations.

Table 1: *Results of LLM rescoring (16-best hypotheses).*

Model	Best λ	SWB	CH
Baseline (Conformer-T)	-	5.3	8.9
BERT-base (uncased)	0.05	5.3	8.8
+scratch	0.15	5.2	8.7
+finetune	0.15	**5.0**	**8.5**
+context	0.10	5.2	8.6
+finetune +context	0.20	**5.0**	**8.5**
RoBERTa-base	0.10	5.2	8.7
+scratch	0.15	5.3	8.6
+finetune	0.20	**5.1**	8.6
+context	0.15	5.2	8.6
+finetune +context	0.20	**5.1**	**8.5**
BERT-large (uncased)	0.05	5.2	8.8
+scratch	0.15	5.2	8.7
+finetune	0.25	**5.1**	8.5
+context	0.10	5.2	8.6
+finetune +context	0.15	**5.1**	**8.4**
RoBERTa-large	0.10	5.1	8.6
+scratch	0.10	5.2	8.6
+finetune	0.25	5.2	**8.5**
+context	0.10	5.1	**8.5**
+finetune +context	0.15	**5.0**	**8.5**
Oracle (16-best)	-	2.6	4.5

Firstly, we note that LLM rescoring with 16-best hypotheses achieves absolute WER reduction of 0.3 for SWB and 0.5

Table 2: *Results of LLM rescoring (100-best hypotheses).*

Model	Best λ	SWB	CH
Baseline (Conformer-T)	-	5.3	8.9
BERT-base (uncased)	0.05	5.3	8.8
+scratch	0.10	5.2	8.7
+finetune	0.15	5.0	8.5
+context	0.10	5.2	8.5
+finetune +context	0.15	**4.9**	**8.4**
RoBERTa-base	0.10	5.2	8.6
+scratch	0.15	5.3	8.6
+finetune	0.15	**5.0**	8.5
+context	0.15	5.2	8.5
+finetune +context	0.15	**5.0**	**8.4**
BERT-large (uncased)	0.10	5.2	8.7
+scratch	0.15	5.2	8.7
+finetune	0.20	**5.1**	8.4
+context	0.10	**5.1**	8.6
+finetune +context	0.15	**5.1**	**8.3**
RoBERTa-large	0.10	5.1	8.6
+scratch	0.10	5.2	8.6
+finetune	0.15	**5.0**	8.5
+context	0.15	5.1	**8.4**
+finetune +context	0.15	**5.0**	**8.4**
Oracle (100-best)	-	1.7	2.8

for CH in the best case. With a larger budget of 100-best hypotheses, additional WER reduction of 0.1 is achieved for both SWB and CH. This is a promising result for LLM-based rescoring with a very competitive ASR baseline.

We can also verify that rescoring with bidirectional LLMs work out-of-the-box using the pretrained parameters. While models trained from scratch are also generally effective (+*scratch*), we observed consistent improvement when the models are rather finetuned from the pretrained parameters (+*finetune*). Therefore, both general-domain pretraining and in-domain finetuning can be essential for LLM rescoring.

When context augmentation is conducted (+*context*), we observed consistent improvement in both pretrained and finetuned LLMs; in fact, the latter achieves the best performance in all cases (+*finetune* +*context*). This result indicates that context augmentation has a complementary benefit with both pretraining and in-domain finetuning.

To further investigate the effect of context augmentation, we've also conducted the experiments with different context sizes, i.e. using the past (left) context only, future (right) context only, shorter ($\times\frac{1}{2}$) context size and longer ($\times 2$) context size. The results are shown in Table 3 (base-size models only).

From this experiment, we could verify that both the left and right context contribute to WER reduction. While longer ($\times 2$) context size improved rescoring in the case of BERT-base, we actually observed competitive performance with shorter ($\times\frac{1}{2}$) context size across all models and sizes (base and large). Therefore, we expect that local context (nearby ~20 words) has the highest impact on LLM rescoring.

In summary, bidirectional (but not unidirectional) LLM rescoring can be effective even on our competitive Conformer-T baseline, and LLMs benefit from all steps of general pretraining, in-domain finetuning and context augmentation.

Table 3: *Context augmentation results with different context sizes (base-size models only, rescored with 100-best hypotheses).*

Model	Context Size		SWB	CH
	# Left	# Right		
BERT-base (uncased) +finetune	40	20	**4.9**	8.4
	40	0	5.0	8.5
	0	20	5.0	8.5
	20	10	5.0	8.4
	80	40	**4.9**	**8.3**
RoBERTa-base +finetune	40	20	**5.0**	**8.4**
	40	0	5.1	**8.4**
	0	20	**5.0**	8.5
	20	10	**5.0**	**8.4**
	80	40	**5.0**	**8.4**

5. Analysis and Discussions

5.1. Lexical Analysis

In this section, we conduct a further analysis on how LLM rescoring improves ASR on our competitive baseline. In prior work, it has been reported that bidirectional LLM rescoring improves ASR in short utterances or at the earlier position of the utterances [9]. It's also reported that bidirectional LLMs are better at judging the linguistic acceptability (e.g. grammaticality) across a wide range of English phenomena [10, 33].

In contrast to prior work, we conduct a simple *lexical analysis* based on the word frequency and error type. To be specific, we first classify each ASR error into the following word frequency classes (*high*, *medium* and *low*) defined based on the transcribed text of 2000 hours Switchboard corpus:

High Frequency: The set of words w whose unigram probability is larger than 0.1 ($P_{\text{UNI}}(w) > 0.1$) on Switchboard. This class includes common pronouns, fillers and function words: e.g. *i*, *you*, *who*, *um*, *well*, *guess*, *could*, *to*, *not*, *when*.

Medium Frequency: The set of words w whose unigram probability is $0.0001 < P_{\text{UNI}}(w) \leq 0.1$ on Switchboard. This class mainly includes the content words: e.g. *listen*, *east*, *women*, *summer*, *brother*, *normal*, *laugh*, *initially*.

Low Frequency: The set of low frequency words that do not belong to either high or medium frequency class. This class includes rare words and named entities: e.g. *firestone*, *realtor*, *threshold*, *worldly*, *unprofitable*, *whatev*, *lehigh*.

The high frequency class contains 138 words and comprises 70.5% of the Switchboard corpus, while the medium frequency class contains 12,049 words and comprises 28.7% of Switchboard. The low frequency class is long-tailed ($\geq$ 35,602 words) but only comprises 0.8% of the corpus.

In addition, we classify each error based on its error type, namely *deletion* and *insertion*. Note that we count a *substitution* error (e.g. *"tried"* $\rightarrow$ *"trade"*) as one deletion error (of *"tried"*) and one insertion error (of *"trade"*) in our analysis.

Based on this classification, we compute the relative error reduction rates against our Conformer-T baseline, separately for each word frequency class and error type.

5.2. Analysis Results

We summarize the results of our lexical analysis in Table 4 (base-size models only). We also report the overall error re-

Table 4: *Relative error reduction rates against our Conformer-T baseline (rescored with 100-best hypotheses, higher ↑ is better). The results are classified based on word frequency (**high**, **medium**, and **low**) and error type (**Del.** for deletion and **Ins.** for insertion). We also report the overall error reduction rate (**Overall**) including both error types and mark the best results in bold.*

	High Freq.			Medium Freq.			Low Freq.		
	Del.	Ins.	Overall	Del.	Ins.	Overall	Del.	Ins.	Overall
BERT-base (uncased)	0.6	−0.5	0.1	2.9	4.1	3.5	2.0	7.9	4.3
+scratch	1.3	−0.4	0.5	4.9	5.8	5.4	3.0	18.7	9.1
+finetune	4.5	1.8	3.2	10.3	13.4	11.8	4.8	22.6	11.7
+context	1.6	3.1	2.3	7.1	10.6	8.8	5.5	15.1	9.2
+finetune +context	5.8	3.1	**4.4**	10.7	14.1	**12.4**	5.5	23.4	**12.4**
RoBERTa-base	2.2	1.4	1.8	7.0	9.4	8.2	4.3	17.9	9.5
+scratch	1.1	0.4	0.8	5.6	3.1	4.4	2.3	31.0	13.4
+finetune	3.5	2.5	3.0	10.0	11.7	10.8	6.0	25.4	**13.5**
+context	0.3	4.7	2.5	7.9	12.3	10.1	7.8	21.4	13.1
+finetune +context	4.7	3.7	**4.2**	10.7	11.4	**11.0**	4.8	25.4	12.7
GPT-2	−5.9	7.0	**0.4**	1.8	−1.1	**0.3**	0.0	7.1	2.8
+finetune	−4.5	5.0	0.1	2.6	−2.0	**0.3**	−2.3	12.3	3.4
+finetune +context	−4.5	4.7	0.0	2.5	−2.0	**0.3**	−1.3	13.5	**4.5**

duction rates (including both deletion and insertion errors)[1] and mark the best result for each word frequency class in bold.

First of all, in general we notice a trade-off between deletion and insertion error reduction, i.e. when deletion error is reduced (correct words are inserted), insertion error tends to increase (incorrect words are inserted). Secondly, there is a significant difference in the results between bidirectional LLMs (BERT and RoBERTa) and unidirectional LLMs (GPT-2), which we discuss separately in the following.

Starting with bidirectional LLMs, we found that the error reduction rates are consistently higher for medium and low frequency word classes. In the low frequency class, the reduction rate for the insertion error is especially high and conspicuous. Therefore, while LLM also improves on high frequency words, LLM's knowledge is most helpful for improving ASR on content words and *removing* incorrect rare words.

Next, if we focus on the results with finetuning (*+finetune*), we can verify that the improvement over scratch training (*+scratch*) is visible in almost all error types and word frequency classes. Therefore, pretraining seems to have a positive effect in general with no (or minimal) side effects.

We can also observe that context augmentation (*+context*) has a benefit similar to finetuning, but there is one interesting difference in the high frequency class: finetuning helps more in reducing deletion errors (i.e. inserting correct words), while context augmentation helps more on the insertion errors (i.e. deleting incorrectly inserted words). This difference seems to have a complementary benefit, and the overall error reduction is most significant (in terms of relative improvement) with the high frequency class when finetuning and context augmentation are combined: e.g. 37.5% improvement ($3.2 \rightarrow 4.4$) for BERT-base with *+finetune* → *+finetune +context*.

Finally, in the unidirectional LLM (GPT-2), we found that the model tends to have an unbalanced effect, improving one type of error but making it worse on the other. For instance, in the high frequency class, insertion errors are significantly reduced in exchange for the increase in deletion errors; this indicates the model is *over-deleting* high frequency words. Such failure modes were not remedied with either finetuning or context augmentation, and we expect that the root of the cause is the lack of bidirectional (left and right) context which is necessary for reliable and effective LLM scoring.

[1]Note that the overall reduction rate is equivalent to the *weighted average* of deletion and insertion error reduction rates.

6. Related Work

As a related work, there have been several attempts to improve the original LLM rescoring method of [9, 10]. For instance, the N-best can be directly rescored based on discriminative training with LLM to optimize for WER reduction [12, 14, 15]. Other works proposed improvements on bidirectional LLM rescoring, e.g. reducing computational cost by predicting the pseudo-log-likelihood based on regression [10, 15] or seeking for the exact log-likelihood through a recursive decomposition of pseudo-log-likelihood [13]. In contrast, the goal of our study is not to propose such improvements but to examine the effect of the original LLM rescoring on a competitive baseline.

Several works also proposed the approach of *knowledge distillation* instead of N-best rescoring to infuse the power of LLMs [34, 35]. One advantage of N-best rescoring is that it requires no modification to the ASR models, allowing for a fast experiment turnover and analysis.

7. Conclusions

In this study, we have re-examined the effect of the fundamental LLM rescoring approach [9, 10] on a competitive Conformer-Transducer baseline and conducted a detailed analysis. Based on our experiments, we have demonstrated consistent improvement in ASR accuracy using bidirectional (but not unidirectional) LLM rescoring. We also observed additional gains from general-domain pretraining, in-domain finetuning and context augmentation when using the bidirectional LLMs.

Lastly, we've conducted a simple lexical-based analysis to examine the effect of LLM rescoring. We showed that error reduction from rescoring can be different across (and characterized by) the word frequency and error type. Based on our analysis, we shed light on how each variant of LLM contributes to WER reduction and explain the failure mode of unidirectional LLMs, being unable to balance both error types.

8. References

[1] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," OpenAI, Tech. Rep., 2019.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[3] 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," *arXiv preprint arXiv:1907.11692*, 2019.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.

[5] I. Tenney, D. Das, and E. Pavlick, "BERT rediscovers the classical NLP pipeline," in *ACL*, 2019.

[6] I. Tenney, P. Xia, B. Chen, A. Wang, A. Poliak, R. T. McCoy, N. Kim, B. V. Durme, S. R. Bowman, D. Das, and E. Pavlick, "What do you learn from context? probing for sentence structure in contextualized word representations," in *ICLR*, 2019.

[7] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller, "Language models as knowledge bases?" in *EMNLP-IJCNLP*, 2019.

[8] S. Ruder, "Recent Advances in Language Model Fine-tuning," http://ruder.io/recent-advances-lm-fine-tuning, 2021.

[9] J. Shin, Y. Lee, and K. Jung, "Effective sentence scoring method using bert for speech recognition," in *ACML*, 2019.

[10] J. Salazar, D. Liang, T. Q. Nguyen, and K. Kirchhoff, "Masked language model scoring," in *ACL*, 2020.

[11] K. Li, Z. Liu, T. He, H. Huang, F. Peng, D. Povey, and S. Khudanpur, "An empirical study of transformer-based neural language model adaptation," *ICASSP*, 2020.

[12] S.-H. Chiu and B. Chen, "Innovative BERT-based reranking language models for speech recognition," in *IEEE SLT*, 2021.

[13] X. Zheng, C. Zhang, and P. C. Woodland, "Adapting GPT, GPT-2 and BERT language models for speech recognition," *ASRU*, 2021.

[14] H. Futami, H. Inaguma, M. Mimura, S. Sakai, and T. Kawahara, "ASR rescoring and confidence estimation with electra," in *ASRU*, 2021.

[15] L. Xu, Y. Gu, J. Kolehmainen, H. Khan, A. Gandhe, A. Rastrow, A. Stolcke, and I. Bulyko, "Rescorebert: Discriminative speech recognition rescoring with bert," in *ICASSP*, 2022.

[16] R. Rosenfeld, "Two decades of statistical language modeling: where do we go from here?" in *Proceedings of the IEEE*, 2000.

[17] T. Mikolov, M. Karafiát, L. Burget, J. H. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH*, 2010.

[18] E. Arisoy, A. Sethy, B. Ramabhadran, and S. Chen, "Bidirectional recurrent neural network language models for automatic speech recognition," in *ICASSP*, 2015.

[19] W. Xiong, L. Wu, J. Zhang, and A. Stolcke, "Session-level language modeling for conversational speech," in *EMNLP*, 2018.

[20] K. Irie, A. Zeyer, R. Schlüter, and H. Ney, "Training language models for long-span cross-sentence evaluation," *ASRU*, 2019.

[21] A. Gulati, C.-C. Chiu, J. Qin, J. Yu, N. Parmar, R. Pang, S. Wang, W. Han, Y. Wu, Y. Zhang, and Z. Zhang, "Conformer: Convolution-augmented transformer for speech recognition," in *INTERSPEECH*, 2020.

[22] Z. Tüske, G. Saon, K. Audhkhasi, and B. Kingsbury, "Single headed attention based sequence-to-sequence model for state-of-the-art results on Switchboard-300," in *INTERSPEECH*, 2020.

[23] A. Wang and K. Cho, "BERT has a mouth, and it must speak: BERT as a Markov random field language model," in *NeuralGen*, 2019.

[24] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush, "Transformers: State-of-the-art natural language processing," in *EMNLP: System Demonstrations*, 2020.

[25] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L.-L. Lim, B. Roomi, and P. Hall, "English conversational telephone speech recognition by humans and machines," in *INTERSPEECH*, 2017.

[26] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *INTERSPEECH*, 2015.

[27] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," *arXiv preprint arXiv:1507.06947*, 2015.

[28] G. Saon, Z. Tüske, D. Bolanos, and B. Kingsbury, "Advancing RNN transducer technology for speech recognition," *arXiv preprint arXiv:2103.09935*, 2021.

[29] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *INTERSPEECH*, 2019.

[30] G. Saon, Z. Tüske, K. Audhkhasi, and B. Kingsbury, "Sequence noise injected training for end-to-end speech recognition," in *ICASSP*, 2019.

[31] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *ICML*, 2013.

[32] G. Saon, Z. Tüske, and K. Audhkhasi, "Alignment-length synchronous decoding for RNN transducer," in *ICASSP*, 2020.

[33] A. Warstadt, A. Parrish, H. Liu, A. Mohananey, W. Peng, S.-F. Wang, and S. R. Bowman, "BLiMP: The benchmark of linguistic minimal pairs for English," *TACL*, 2020.

[34] H. Futami, H. Inaguma, S. Ueno, M. Mimura, S. Sakai, and T. Kawahara, "Distilling the knowledge of BERT for sequence-to-sequence ASR," in *INTERSPEECH*, 2020.

[35] Y. Kubo, S. Karita, and M. Bacchiani, "Knowledge transfer from large-scale pretrained language models to end-to-end speech recognizers," in *ICASSP*, 2022.