CORRECTION OF AUTOMATIC SPEECH RECOGNITION WITH TRANSFORMER SEQUENCE-TO-SEQUENCE MODEL

Oleksii Hrinchuk[12*] *Mariya Popova*[23*] *Boris Ginsburg*[2]

[1]Moscow Institute of Physics and Technology, Moscow, Russia
[2]NVIDIA, Santa Clara, CA, USA
[3]Carnegie Mellon University, Pittsburgh, PA, USA
{aleksey.grinchuk@phystech.edu, mpopova@andrew.cmu.edu, bginsburg@nvidia.com}

ABSTRACT

In this work, we introduce a simple yet efficient post-processing model for automatic speech recognition. Our model has Transformer-based encoder-decoder architecture which "translates" acoustic model output into grammatically and semantically correct text. We investigate different strategies for regularizing and optimizing the model and show that extensive data augmentation and the initialization with pre-trained weights are required to achieve good performance. On the LibriSpeech benchmark, our method demonstrates significant improvement in word error rate over the baseline acoustic model with greedy decoding, especially on much noisier dev-other and test-other portions of the evaluation dataset. Our model also outperforms baseline with 6-gram language model re-scoring and approaches the performance of re-scoring with Transformer-XL neural language model.

Index Terms— speech recognition, spelling correction, pre-trained language models

1. INTRODUCTION

In recent years, automatic speech recognition (ASR) research has been dominated by end-to-end (E2E) models [1, 2, 3] which outperformed conventional hybrid systems relying on Hidden Markov Models [4, 5]. In contrast to prior work, which required training several independent components (acoustic, pronunciation, and language models) and had many degrees of complexity, E2E models are faster and easier to implement, train, and deploy.

To enhance the speech recognition accuracy, ASR models are often augmented with independently trained language models that re-score the list of n-best hypotheses. The use of the external language model induces a natural trade-off between the speech recognition speed and accuracy. While simple N-gram language models (e.g., KenLM [6]) are extremely fast, they can not achieve the same level of performance as heavier and more powerful neural language models, such as Transformers [7, 8, 9].

Language model re-scoring effectively expands the search space of speech recognition candidates; however, it can barely help when the ground truth word was assigned a low score by erroneous ASR model. Traditional left-to-right language models are also prone to error accumulation: if some word at the beginning of the decoded speech is misrecognized, it will affect the scores of all succeeding words by providing them with incorrect context. To address these problems, we propose to train a conditional language model that corrects the errors made by the system operating similar to neural machine translation (NMT) [10, 11] by "translating" corrupted ASR output into the correct language.

There is a plethora of prior work on correcting ASR systems output, and we refer the reader to [12] for a detailed overview. Most closely to our work, [13] propose to train a spelling correction model based on RNN with attention [14] to correct the output of Listen, Attend and Spell (LAS) model. In contrast to this work, our model is based on Transformer architecture [15] and does not require a complementary text-to-speech model for training data generation.

Transformers used for NMT are usually trained on millions of parallel sentences and tend to easily overfit if the data is scarce, which is the case we have. To solve this problem, we propose two self-complementary regularization techniques. First, we augment training data with the perturbed outputs of several ASR models trained on K-fold partitions of the training dataset. Second, we initialize both encoder and decoder with the weights of pre-trained BERT [16] language model, which was shown to be efficient for transfer learning in various natural language processing tasks.

We evaluate the proposed approach on LibriSpeech dataset [17] and use Jasper DR 10x5 [7] as our baseline ASR module. Our correction model, when applied to the greedy output of Jasper ASR, outperforms both the baseline and re-scoring with 6-gram KenLM language model and almost matches the performance of re-scoring with more powerful Transformer-XL language model.

*Equal contribution, work done during an internship at NVIDIA

2. MODEL

2.1. Speech recognition baseline model

As our baseline ASR model we use Jasper [7], a deep convolutional E2E model. Jasper takes as input mel-filter bank features calculated from 20ms windows with a 10ms overlap and maps them to a probability distribution over characters per frame. The model is trained with Connectionist Temporal Classification (CTC) loss [18]. In particular, we build on Jasper-DR-10x5, which consists of 10 blocks of 5 sub-blocks (1-D convolution, batch norm, ReLU, dropout) where the output of each block is added to the inputs of all following blocks similar to DenseNet [19].

The baseline Jasper model is trained with the Novograd [20] optimizer and implemented in PyTorch within NeMo toolkit [21].

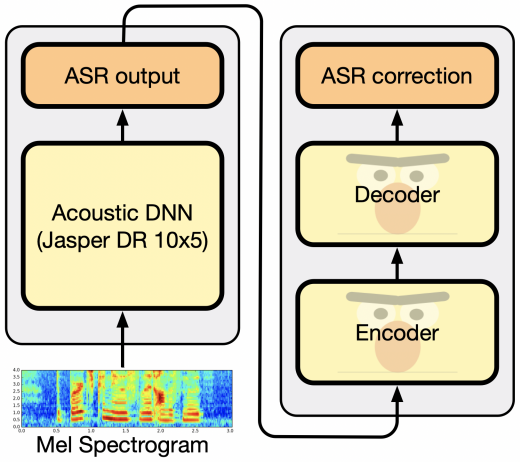

Fig. 1. ASR correction model based on Transformer encoder-decoder architecture.

2.2. Language models used for re-scoring

A language model (LM) estimates the joint probability of a text corpus $(x_1, \ldots, x_T)$ by factorizing it with a chain rule $P(x_1, \ldots, x_T) = \prod_{t=1}^{T} P(x_t|x_1, \ldots, x_{t-1})$ and sequentially modeling each conditional term in the product. To simplify modeling, it is often assumed that the context size (a number of preceding words) necessary to predict each word x_t in the corpus is limited to N: $P(x_t|x_1, \ldots, x_{t-1}) \approx P(x_t|x_{t-N}, \ldots, x_{t-1})$. This approximation is commonly referred to as N-gram LM.

Following the original Jasper paper [7], we are considering two different LMs: 6-gram KenLM [6] and Transformer-XL [8] with the full sentence as the context. For generation, a beam search with the width 2048 is used where each hypothesis is evaluated with shallow fusion of acoustic and language models:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} \left[\log P_{ASR}(\mathbf{y}|\mathbf{x}) + \lambda \log P_{LM}(\mathbf{y})\right] \quad (1)$$

It is worth noting that Transformer-XL does not replace KenLM but complements it. Beam search in hypothesis formation is governed by the joint score of ASR and KenLM, and the resulting beams are additionally re-scored with Transformer-XL in a single forward pass. Using Transformer-XL instead of KenLM all the way along is too expensive due to much slower inference of the former.

2.3. ASR correction model

The proposed model (Figure 1) has Transformer encoder-decoder architecture [15] commonly used for neural machine translation. We denote the number of layers in encoder and decoder as L, the hidden size as H, and the number of self-attention heads as A. Similar to prior work [15, 16], the fully-connected inner-layer dimensionality is set to $4H$. Dropout with probability P_{drop} is applied after the embedding layer, after each sub-layer before the residual connection, and after multi-head dot-product attention.

We consider two options for initializing the model weights: random initialization and using the weights of pre-trained BERT [16]. Since BERT has the same architecture as Transformer encoder, its parameters can be straightforwardly used for encoder initialization. In order to initialize the decoder, which has an additional encoder-decoder attention block in each layer, we duplicate and use the parameters of the corresponding self-attention block.

3. EXPERIMENTS

3.1. Dataset

We conduct our experiments on LibriSpeech [17] benchmark. Librispeech training dataset consists of three parts — train-clean-100, train-clean-360, and train-clean-500, which together provide 960 hours of transcribed speech or around 281K training sentences. For evaluation, LibriSpeech provides 2 development datasets (dev-clean and dev-other) and 2 test datasets (test-clean and test other). We found that even baseline models made only a few mistakes on dev-clean and selected the checkpoint with the lowest WER on dev-other for evaluation.

To generate training data for our Transformer ASR correction model, we split all training data into 10 folds and trained 10 different Jasper models in a cross-validation manner: each model was trained on 9 folds and used to generate greedy ASR predictions for the remaining 10th fold. Then, we concatenated all resulting 10 folds and used Jasper greedy predictions as the source side of our parallel corpora with ground truth transcripts as the target side.

However, we did not manage to considerably improve upon Jasper greedy when training the Transformer on resulting 281K training sentences because of extreme overfitting. To augment our training dataset, we used two techniques:

- We took pre-trained Jasper model and enabled dropout during inference on training data. This procedure was repeated multiple times with different random seeds.
- We perturbed training data with Cutout [22] by randomly cutting small rectangles out of the spectrogram, which essentially drops complete phonemes or words and mel frequency channels.

After the augmentation, deduplication[1], and removal of sentence pairs with WER greater than 0.5, we ended up with approximately 2.5M of training examples.

The ablation study of the proposed data augmentation techniques is presented in Table 1. In our experiments, training on the full dataset of sentences generated with enabled dropout and cutout was much more efficient; thus, we stick to it as our training dataset in all subsequent experiments. We also experimented with using top-k beams obtained with beam search but found that the resulting sentences lacked in diversity, often differing in a few characters only.

Dataset	Size	Dev		Test	
		clean	other	clean	other
Jasper greedy	–	3.64	11.89	3.86	11.95
10-fold	281K	3.55	9.70	3.81	9.96
+ cutout	1.1M	3.35	9.56	3.75	9.79
+ dropout	1.7M	3.31	9.20	3.81	9.52
+ both	2.5M	3.26	9.01	3.54	9.34

Table 1. Ablative study of data augmentation techniques.

Recently, several promising data augmentation techniques were introduced for both NLP and ASR, such as adding noise to the beams [23, 24] and SpecAugment [25] which are also applicable in our case. However, it goes beyond the scope of this paper, and we leave it for future work.

3.2. Training

All models used in our experiments are Transformers with parameters ($H = 768, A = 12, P_{\text{drop}} = 0.25, L = 12$). We train them with NovoGrad [20] optimizer (lr $= 0.001, \beta_1 = 0.95, \beta_2 = 0.25$) for a maximum of 300K steps with polynomial learning rate decay on a batches of 32K source and target tokens. For our vocabulary we adopted 30K WordPieces [26] used in BERT [16] so we could straightforwardly transfer its pre-trained weights. According to [15], we also used label smoothing of 0.1 for regularization. Each model was trained on a single DGX-1 machine with 8 NVIDIA V100 GPUs. Models were implemented in PyTorch within NeMo toolkit[2].

[1] As we applied the described above perturbations multiple times to each sentence, some of the obtained training samples appeared to be identical.

3.3. Initialization

Next, we experiment with various architectures and initialization schemes. Specifically, we either initialize all weights randomly (`rand`) from $\mathcal{N}(0, 0.02)$ or transfer weights from the pre-trained `bert-base-uncased` model (`BERT`). Table 2 depicts the performance of different configurations.

Model		Dev		Test	
encoder	decoder	clean	other	clean	other
Jasper greedy		3.64	11.89	3.86	11.95
Jasper + 6-gram		2.89	9.53	3.34	9.62
Jasper + TXL		2.68	8.62	2.95	8.79
rand	rand	3.92	10.30	4.22	10.63
rand	BERT	3.89	9.92	4.19	10.29
BERT	rand	3.26	9.01	3.54	9.34
BERT	BERT	3.18	8.98	3.50	9.27

Table 2. Performance of our model with different initialization schemes in comparison to the baselines. Jasper results are taken from the original paper [7].

Models with randomly initialized encoder improve upon the results of Jasper greedy on "other" portions of evaluation datasets; however, their correction harms the performance on the "clean" portion. Adding BERT-initialized decoder achieves slightly better results, but it still lags behind the baseline Jasper with LM re-scoring.

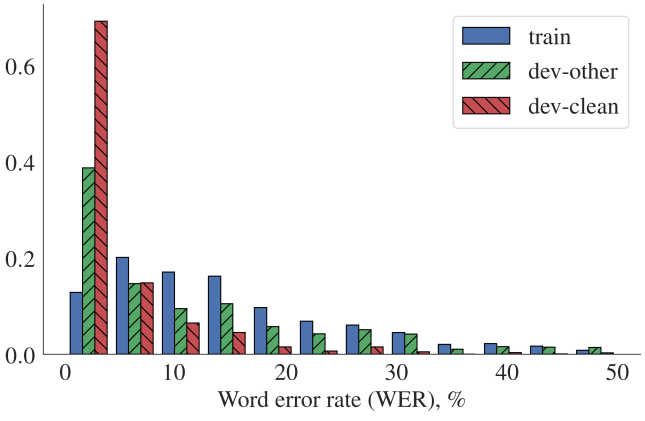

Fig. 2. WER distribution of training and evaluation datasets.

Models with BERT-initialized encoder are strictly better than both Jasper greedy and models with encoder initialized randomly. They outperform Jasper with 6-gram KenLM on

[2] `https://github.com/nvidia/nemo`

Model	Example 1	Example 2	Example 3
Ground truth	pierre looked at him in surprise	i've gained fifteen pounds and	and how about little avice caro
Greedy	pure locat e ham in a surprise	afgain fifteen pounds and	and hawbout little ov his carrow
+ 6-gram	pure locate him in surprise	again fifteen pounds and	and __ about little of his care
+ TXL	pure locate him in surprise	again fifteen pounds and	and __ but little of his care
Ours	**pierre looked at him** in surprise	**i've gained** fifteen pounds and	and **how about** little of his care

Table 3. Outputs produced by different models. Both greedy decoding and re-scoring with external LMs fail to recognize the beginning of the speech which is poorly decoded by acoustic model and has little or no context for LM. Our model succeeds by leveraging the context of corrupted yet complete decoded excerpt.

Model	Example
Ground truth	one day the traitor fled with a teapot and a basketful of cold victuals
Greedy	one day the trade of fled with he teappot and a basketful of cold victures
+ 6-gram	one day the trade of fled with the tea pot and a basketful of cold pictures
+ TXL	one day the trade of fled with the tea pot and a basketful of cold victores
Ours	one day the trader fled with **a teapot** and a basketful of cold **victuals**

Table 4. Combination of acoustic and language models fails to generate the subject of the sentence which leads to further errors. While our ASR correction model does not manage to fully reconstruct the ground truth transcript, its output is coherent English with the last word successfully corrected.

"other" portions of evaluation datasets and approach the accuracy of the Jasper with Transformer-XL re-scoring. The best performance is achieved by the model with both encoder and decoder initialized with pre-trained BERT.

Interestingly, our ASR correction model considerably pushes forward the performance on much noisier "other" evaluation datasets and only moderately improves the results on "clean". This can be explained by the slight domain mismatch in our training data and "clean" evaluation datasets. Our data was collected with the models which achieve around 14% WER on average (or even higher, if dropout is on during generation) and does not contain many "clean" examples, which usually have much lower WER. Figure 2 shows that the distribution of WER in training data is indeed much closer to dev-other than to dev-clean.

3.4. Analysis of corrected examples

To conduct a qualitative analysis of ASR corrections produced by our best model with BERT-initialized encoder and decoder, we examined the examples on which it successfully corrects the output of Jasper greedy.

Table 3, which depicts the excerpts with the largest difference in WER between greedy and corrected ASR outputs, reveals an interesting pattern. Both greedy decoding and re-scoring with external LMs make a mistake at the very beginning of the transcript. If it is poorly decoded by acoustic model and has little or no context for LM, the scores used for evaluating beams in Equation (1) are simply unreliable. The mistakenly generated context might also negatively affect the succeeding left-to-right LM scores leading to even more errors (Table 4). Our model, on the other hand, successfully corrects ASR output by leveraging the bidirectional context provided by corrupted yet complete decoded excerpt.

4. CONCLUSION

In this work, we investigated the use of Transformer-based encoder-decoder architecture for the correction of ASR systems output. The proposed ASR output correction technique is capable of "translating" the erroneous output of the acoustic model into grammatically and semantically correct text.

The proposed approach enhances the acoustic model accuracy by a margin comparable to shallow fusion and re-scoring with external language models. Analysis of corrected examples demonstrated that our model works in scenarios when the scores produced by both acoustic and external language models are not reliable.

To overcome the problem of extreme overfitting on the relatively small training dataset, we proposed several data augmentation and model initialization techniques, i.e., enabling dropout and Cutout [22] during acoustic model inference and initializing both encoder and decoder with the parameters of pre-trained BERT [16]. We also performed a series of ablation studies showing that both data augmentation and model initialization have a significant impact on model performance.

5. REFERENCES

[1] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP*, 2016.

[2] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *ICASSP*. IEEE, 2016.

[3] Y. Zhang, W. Chan, and N. Jaitly, "Very deep convolutional networks for end-to-end speech recognition," in *ICASSP*. IEEE, 2017, pp. 4845–4849.

[4] Y. Bengio, R. De Mori, G. Flammia, and R. Kompe, "Global optimization of a neural network-hidden markov model hybrid," in *IJCNN-91-Seattle*. IEEE, 1991, vol. 2, pp. 789–794.

[5] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury, et al., "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal processing magazine*, vol. 29, 2012.

[6] K. Heafield, "Kenlm: Faster and smaller language model queries," in *Proceedings of the sixth workshop on statistical machine translation*. Association for Computational Linguistics, 2011, pp. 187–197.

[7] 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," *Interspeech*, 2019.

[8] Z. Dai, Z. Yang, Y. Yang, W. W. Cohen, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," *arXiv preprint arXiv:1901.02860*, 2019.

[9] A. Baevski and M. Auli, "Adaptive input representations for neural language modeling," *ICLR*, 2019.

[10] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *NIPS*, 2014.

[11] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *EMNLP*, 2014.

[12] R. Errattahi, A. El Hannani, and H. Ouahmane, "Automatic speech recognition errors detection and correction: A review," *Procedia Computer Science*, 2018.

[13] J. Guo, T. N. Sainath, and R. J. Weiss, "A spelling correction model for end-to-end speech recognition," in *ICASSP*. IEEE, 2019, pp. 5651–5655.

[14] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

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

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

[17] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*. IEEE, 2015, pp. 5206–5210.

[18] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006.

[19] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[20] B. Ginsburg, P. Castonguay, O. Hrinchuk, O. Kuchaiev, V. Lavrukhin, R. Leary, J. Li, H. Nguyen, and J. M. Cohen, "Stochastic gradient methods with layer-wise adaptive moments for training of deep networks," *arXiv preprint arXiv:1905.11286*, 2019.

[21] O. Kuchaiev, J. Li, H. Nguyen, O. Hrinchuk, R. Leary, B. Ginsburg, S. Kriman, S. Beliaev, V. Lavrukhin, J. Cook, et al., "Nemo: a toolkit for building ai applications using neural modules," *arXiv preprint arXiv:1909.09577*, 2019.

[22] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[23] S. Edunov, M. Ott, M. Auli, and D. Grangier, "Understanding back-translation at scale," *EMNLP*, 2018.

[24] W. Zhang, Y. Feng, F. Meng, D. You, and Q. Liu, "Bridging the gap between training and inference for neural machine translation," *ACL*, 2019.

[25] 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," *arXiv preprint arXiv:1904.08779*, 2019.

[26] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al., "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.