



Investigation of Transformer based Spelling Correction Model for CTC-based End-to-End Mandarin Speech Recognition

Shiliang Zhang, Ming Lei, Zhijie Yan

Machine Intelligence Technology, Alibaba Group

{sly.zsl, lm86501, zhijie.yzj}@alibaba-inc.com

Abstract

Connectionist Temporal Classification (CTC) based end-to-end speech recognition system usually need to incorporate an external language model by using WFST-based decoding in order to achieve promising results. This is more essential to Mandarin speech recognition since it owns a special phenomenon, namely *homophone*, which causes a lot of substitution errors. The linguistic information introduced by language model is somehow helpful to distinguish these substitution errors. In this work, we propose a transformer based spelling correction model to automatically correct errors, especially the substitution errors, made by CTC-based Mandarin speech recognition system. Specifically, we investigate to use the recognition results generated by CTC-based systems as input and the ground-truth transcriptions as output to train a transformer with encoder-decoder architecture, which is much similar to machine translation. Experimental results in a 20,000 hours Mandarin speech recognition task show that the proposed spelling correction model can achieve a CER of 3.41%, which results in 22.9% and 53.2% relative improvement compared to the baseline CTC-based systems decoded with and without language model, respectively.

Index Terms: speech recognition, spelling correction, CTC, End-to-End, Transformer

1. Introduction

Conventional hybrid DNN-HMM based speech recognition system usually consists of acoustic, pronunciation and language models. These components are trained separately, each with a different objective, and then combined together during model inference. Recent works in this area attempt to rectify this disjoint training problem and simplify the training process by building speech recognition system in the so-called end-to-end framework [1, 2, 3, 4, 5, 6, 7, 8, 9]. Two popular approaches are the Connectionist Temporal Classification (CTC) [10] and attention-based encoder-decoder framework [11]. Both methods regard speech recognition as a sequence-to-sequence mapping problem and address the problem of variable-length input and output sequences. CTC uses intermediate label representation allowing repetitions of labels and occurrences of blank label to identify less informative frames, which enables CTC-based acoustic models to automatically learn the alignments between speech frames and target labels. On the other hand, attention based models use an attention mechanism to perform alignment between acoustic frames and recognized symbols. Both methods do not require frame-level training targets, which simplifies the training process of speech recognition system.

CTC assumes that the label outputs are conditionally independent of each other, which can be seen as an acoustic-only model. Although CTC-based models can directly generate the recognition results by using the greedy search decoding [10], it's better to incorporate an external language model at

the character or word level by using the WFST-based decoding [12, 13]. On the other hand, attention-based models with *encoder-decoder* architecture can jointly learn acoustic, pronunciation and language models. As a result, it is widely observed that attention-based models will achieve better performance than CTC-based models when decoded without external language model [7]. However, the language model component in attention-based models is only trained on transcribed audio-text pairs. Further improvements can achieve by incorporating an external language model during model inference [14].

Mandarin is a tonal language with a special phenomenon, namely *homophone*, which many Chinese characters share the same pronunciation. As a result, the substitution errors are the dominant error made by Mandarin speech recognition system [15]. These substitution errors require linguistic information to distinguish effectively. Thereby, the language model is essential to CTC-based models for Mandarin speech recognition. As shown in [13], the performance gap between CTC-based models decoded with and without external language model is huge. However, the language model accompanied with CTC-based acoustic models is usually the n-gram language model, which has limited history context information. Further improvement can be obtained by using N-best rescoring with an RNN-LM [16, 17].

In this work, we propose a transformer [18] based spelling correction model to automatically correct some errors made by the CTC-based speech recognition system. Specifically, we investigate using the recognition results generated by CTC-based systems as input and the ground-truth transcriptions as output to train a transformer with encoder-decoder architecture, which is similar to machine translation. During inference, the spelling correction model takes the preliminary recognition results as input and generates the final results with greedy search. We have investigated various CTC-based systems as front-end: different acoustic modeling units (syllable, character-2k, character-4k, character-6k), different optimization criteria (CTC, CTC-sMBR) and decoding methods (greedy search, WFST search). Moreover, we propose to use the SGDR [19] optimization based model training and N-best lists based data expansion methods to further improve the performance.

We have evaluated our proposed approach on a 20,000 hours Mandarin speech recognition task, which consists of about 20 millions paired sentences. Our results show that the proposed spelling correction models can improve the performance of CTC system with greedy search from 7.28% to 4.89% in character error rate (CER). And it can further improve to 4.21% by extending N-best lists as training data. As a comparison, the performance of a well-trained CTC-sMBR system using WFST-based decoding with external word-level language model is 4.42%. Moreover, by jointly using character based acoustic modeling units, DFSMN-CTC-sMBR acoustic model, WFST-based decoding and N-best data expansion, the proposed

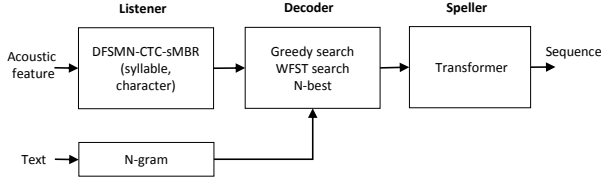


Figure 1: Overall system of the proposed approach.

spelling correction models can achieve a CER of 3.41%, which results in a 22.9% relative improvement. Our analysis show that the transformer based spelling correction model can significantly reduce substitution errors in recognition results, due to it can utilize the sentence-level linguistic information.

2. Related Works

Automatic correction of recognition errors is crucial not only to improve the performance of ASR system but also to avoid the propagation of errors to the post process (e.g. machine translation, natural language processing). [20] presents an overview of previous work on error correction for ASR. However, most of researches were limited to the detection [21, 22, 23] and just few researches addressed the correction process of ASR errors. In [24], it built an ASR errors detector and corrector using co-occurrence analysis. [25] proposed a post-editing ASR errors correction method based on Microsoft N-gram dataset. More recently, work in [26, 27] proposed to use the attention based sequence-to-sequence model to automatically correct the ASR errors, which is much similar to our work.

3. Our Approach

Figure 1 demonstrates the overall system of the proposed approach, which consists of three components: *listener*, *decoder* and *speller*. For listener, we use the DFSMN-CTC-sMBR [15] based acoustic model. As to decoder, we compare the greedy search [10] and WFST search [12] based decoding strategies to generate preliminary recognition results given static sequence of probabilities generated by the listener. Moreover, we also investigate how to extend the diversity of preliminary recognition results with candidate N-best. Finally, the outputs generated by the decoder are used to train a transformer [18] based speller.

3.1. Listener

3.1.1. DFSMN-CTC-sMBR

DFSMN [28] is an improved FSMN [29] structure that enables to build extreme deep architecture by introducing skip connections. The key element in DFSMN is the learnable FIR-like memory blocks, which are used to encode long context information into fixed-size representation. As a result, DFSMN is able to model the long-term dependency in sequential signals while without using recurrent feedback. The operation in ℓ -th memory block takes the following form:

$$\mathbf{m}_t^\ell = \mathbf{m}_t^{\ell-1} + \mathbf{p}_t^\ell + \sum_{i=0}^{N_1^\ell} \mathbf{a}_i^\ell \odot \mathbf{p}_{t-s_1*i}^\ell + \sum_{j=1}^{N_2^\ell} \mathbf{c}_j^\ell \odot \mathbf{p}_{t+s_2*j}^\ell \quad (1)$$

Here, \mathbf{p}_t^ℓ denote the outputs of the linear projection layer and \mathbf{m}_t^ℓ denotes the output of the memory block. N_1^ℓ and N_2^ℓ denotes the look-back order and lookahead order of the memory

block, respectively. s_1 is the stride factor of look-back filter and s_2 is the stride of lookahead filter.

Connectionist temporal classification (CTC) [10] is a loss function for sequence labeling problems that converts the sequence of labels with timing information into the shorter sequence of labels by removing timing and alignment information. The main idea is to introduce the additional CTC blank (−) label during training, and then remove the blank labels and merging repeating labels to obtain the unique corresponding sequence during decoding. For a set of target labels, Ω , and its extended CTC target set is defined as $\bar{\Omega} = \Omega \cup \{-\}$. Given an input sequence \mathbf{x} and its corresponding output label sequence \mathbf{y} . The CTC path, π , is defined as a sequence over $\bar{\Omega}$, $\pi \in \bar{\Omega}^T$, where T is the length of the input sequence \mathbf{x} . The label sequence \mathbf{y} can be represented by a set of all possible CTC paths, $\Phi(\mathbf{y})$, that are mapped to \mathbf{y} with a sequence to sequence mapping function \mathcal{F} , $\mathbf{y} = \mathcal{F}(\Phi(\mathbf{y}))$. Thereby, the log-likelihood of reference label sequence \mathbf{y} given the input \mathbf{x} can be calculated as an aggregation of the probabilities of all possible CTC paths:

$$p(\mathbf{y}|\mathbf{x}) = \sum_{\pi \in \Phi(\mathbf{y})} p(\pi|\mathbf{x}) \quad (2)$$

Model training can then be carried out by minimizing the negative log-likelihood. Furthermore, CTC trained acoustic model can be further optimized with sequence-level discriminative training criteria such as state-level minimum Bayes risk (sMBR) criterion [30, 28].

3.1.2. Acoustic modeling units

In [15], it have investigated the performance of DFSMN-CTC-sMBR acoustic models with CI-IF, CD-IF, syllable and hybrid Character-Syllable as modeling units for Mandarin speech recognition. Experimental results suggest that the *hybrid Character-Syllable* modeling units, which mixed the high frequency Chinese characters and syllables, is the best choice for Mandarin speech recognition. For *hybrid Character-Syllable*, the low frequency characters are mapped into the syllables to deal with the OOV problem. In this work, instead of mapping the low frequency characters into syllables, we propose to map them into high frequency characters with the same pronunciation. As a result, we come up with a pure Chinese characters based modeling units without OOV for Mandarin speech recognition. Specifically, we keep the top 2000, 4000 and 6000 Chinese characters as acoustic modeling units, denoted as *char-2k*, *char-4k* and *char-6k* respectively. The coverages is 95.58%, 99.54% and 99.86% in our text dataset, respectively.

3.2. Decoder

3.2.1. Greedy search

For greedy search [10] based decoding, the most likely symbol at each time step is chosen as the output. The best CTC-path can be generated as followings:

$$\pi^* = \arg \max_{\pi} \prod_{t=1}^N P_{AM}(\pi_t|\mathbf{x}) \quad (3)$$

Furthermore, the CTC-path can be mapped to the token sequence by using the mapping function \mathcal{F} .

$$\mathbf{y} = \mathcal{F}(\pi^*) \quad (4)$$

For Chinese character units based CTC acoustic model, the token sequence is the final recognition results. For syllable or other modeling units, it still need another mapping function.

Table 1: Model architecture of the Transformers used in this work.

Transformer	N	d_{model}	d_{ff}	h	d_k	d_v
small	3	512	2048	4	512	512
big	6	512	2048	8	512	512

3.2.2. WFST search

The WFST-based [31] decoding method proposed in [12] that enables efficient incorporation of lexicons and language models into CTC decoding. The search graph is built by composing the language model WFST G , lexicon WFST L and token WFST T . The token WFST T maps a sequence of frame-level CTC labels to a single lexicon unit. The overall order of the FST operations is:

$$S = T \circ \min(\det(L \circ G)) \quad (5)$$

where \circ , \det and \min denote composition, determinization and minimization respectively. The search graph S encodes the mapping from a sequence of CTC labels emitted on speech frames to the final transcription. The best decoding path can then be exported from the search graph S using the beam search.

3.2.3. N-best data expansion

For both greedy search and WFST search based decoding, we usually take the best path as recognition result. In our work, we find the diversity of training data is important to the performance of spelling correction model. Thereby, we investigate to extend the diversity of training data using the N-best lists. For WFST-based decoding, we can easily get the top N paths from the decoding lattice. As to greedy search based decoding, we propose a threshold-based path retention method. At each time-step, in addition to retaining the token with the highest posterior probability (p_1), we judge whether to retain the second token (posterior probability, p_2) based on two thresholds ($upper_th, lower_th$). if $lower_th < p_1 < upper_th$ and $lower_th < p_2$, we will keep both tokens. Otherwise, we only keep the top one token. Based on different thresholds, we will generate different CTC paths that can further mapped to token sequence with Eq.4.

In our work, we also investigate to extend the training data using data augmentation with the text-only data. We try to add insertion, deletion and substitution errors to the original text based on a probability distribution. Unfortunately, this method doesn't work well. We suspect that constructed errors can't really simulate the types of error produced by the acoustic model.

3.3. Speller

For spelling correction model (*speller*), we use the *Transformer* with encode-decoder architecture, which is much similar to machine translation model in [18]. The preliminary recognition results generated by the front-end CTC-based acoustic models with different decoding and data expansion methods are used as input and the ground-truth transcriptions are used as output to train the speller. We use the OpenNMT toolkit [32] to train the Transformer based speller with default setting. Specially, we have compared two Transformer architectures, denoted as *small* and *big*, and the detailed configurations are as shown in Table 1.

Table 2: CER (%) of CTC based ASR systems with various acoustic modeling units and decoding methods (Greedy search Vs. WFST search).

Exp	Modeling units	Criterion	Greedy Search	WFST Search
1	syllable	CTC	-	5.55
2	char-2k	CTC	11.93	5.21
3	char-4k	CTC	7.28	5.20
4		+SMBR	8.48	4.42
5	char-6k	CTC	7.33	5.25

Table 3: Performance of greedy search CTC with speller. (1M: 1 million sentences.)

Exp	Modeling units	Transformer	Data	CER%
1	syllable	small	1M	8.17
			20M	6.40
2	char-2k	small	20M	6.25
3	char-4k	small	20M	5.70
4	char-4k	big	20M	5.55
5	char-6k	big	20M	5.65

4. Experiments

4.1. Experimental setup

We conduct our experiments on a large Mandarin speech recognition task that consists of about 20,000 hours training data with about 20 million sentences. A test set contains about 10 hours data is used to evaluate the performance of all models. Acoustic feature used for all experiments are 80-dimensional log-mel filter-bank (FBK) energies computed on 25ms window with 10ms shift. We stack the consecutive frames within a context window of 5 (2+1+2) to produce the 400-dimensional features and then down-sample the inputs frame rate to 30ms. For WFST-based decoding, a pruned trigram language model trained with the text data is used. Evaluations are performed in term of character error rate (CER). For all experiments, we use the same DFSMN architecture as in [33]. CTC-based acoustic model is trained in a distributed manner using 16 GPUs and the Transformer based speller is trained using 2 GPUs.

4.2. CTC baseline system

The performance of various baseline CTC-based ASR systems with different acoustic modeling units and decoding methods are shown in Table 2. CTC-based models decoded with WFST search perform much better than greedy search, which indicates the importance of linguistic information. Chinese character modeling units based CTC models with different numbers of characters (2k, 4k, 6k) can achieve similar performance when using WFST-based decoding, and all can outperform the syllable based CTC model. However, when decoded with greedy search, char-4k and char-6k based CTC models significantly outperform the char-2k based CTC model. This experimental result also indicates the importance of linguistic information. As to sMBR training, we find that it improves the performance when using WFST based decoding while hurt the performance when using greedy search based decoding. This is due to the mismatch between training and decoding, since sMBR based training uses the WFST-based decoding to generate the training lattices.

Table 4: Performance of speller trained with data generated by greedy search CTC and threshold-based data expansion.

Training data	Steps/Pass	Pass1	Pass2	Pass3	Pass4
D1	100000	5.55	5.18	5.02	4.89
D1-3	200000	4.73	4.68	4.46	4.36
D1-6	400000	4.62	4.38	4.28	4.21

Table 5: Performance of spellers trained with data generated by WFST search CTC and N-best data expansion.

Training data	Steps/Pass	Pass1	Pass2	Pass3	Pass4
N-best(1)	100000	4.14	4.01	3.98	3.91
N-best(5)	250000	3.79	3.69	3.60	3.54
N-best(10)	350000	3.72	3.61	3.50	3.41

4.3. Greedy search CTC with speller

First, we evaluate the performance of spellers trained with the output of baseline CTC models in Table 2 using greedy search based decoding. Comparison of the experimental results of *exp2* and *exp3* in Table 3 and Table 2 shows that good preliminary recognition result will lead to better final result. Moreover, the amount of training data is essential to the performance. As shown in Table 3, increasing the training data from 1 million (1M) sentences to 20M sentences will result in more than 20% relative improvement. Thereby, we also investigate how to extend the training data in this work. Since the *big* Transformer based speller performs better than the *small* one, we will use the *big* one in our following experiments.

In Sec.3.2.3, we have introduced the threshold-based data expansion method for CTC-based acoustic model with greedy search. In this experiment, we use the baseline models in Table 3 with different thresholds to generate various training data, denoted as D(AM *upper_th lower_th*). Specially, we use the CTC model (*exp3*) and CTC-sMBR model (*exp4*) to generate six datasets, denoted as D1 (CTC 1.0 1.0), D2(CTC 0.5 0.1), D3(CTC 0.6 0.1), D4(CTC-sMBR 1.0 1.0), D5(CTC-sMBR 0.5 0.3), D6(CTC-sMBR 0.6 0.3). Different data configurations and experimental results are as shown in Table 4. Inspired by the SGDR [19], we propose to train the speller with 4 passes and reset the learning rate after each pass. The training steps of each pass is determined by the amount of training data. For model inference, we use the results of CTC-based model with greedy search as input and generate the final recognition result by also using greedy search. Results show that this optimization method will achieve a better convergence performance. More importantly, data expansion will significantly improve the performance of speller even using the training data generated by CTC-sMBR model. As a result, we can achieve a character error rate (CER) of 4.21%, which is 42.17% relative improvement compared to the baseline greedy search decoded CTC model (*exp3* in Table 4.2). It can also outperform the CTC-sMBR model decoded with external language model using the WFST-based decoding.

4.4. WFST search CTC with speller

In this experiment, we use the DFSMN-CTC-sMBR listener (*exp4* in Table 4.2) with WFST-based decoder to generate the preliminary recognition results, which are then paired with the ground-truth transcriptions to train the speller. We also extend the training data with N-best lists in decoding lattice. Specially, we keep the top 1, 5 and 10 paths that result in three

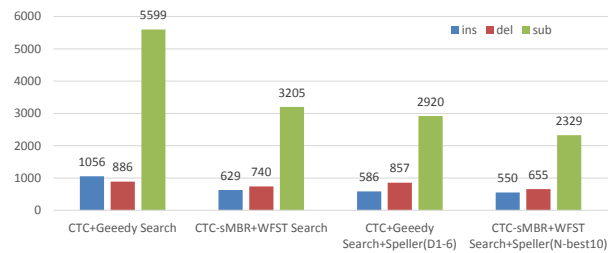


Figure 2: Error analysis of various systems.

DFSMN-CTC-sMBR + WFST-based Decoder	+ Speller
我们一直是按双轨制在推进一个 鬼叫公共鬼另外一个轨叫商业鬼 (We have been pushing on a two-track system one ghost is called the public ghost and the other is called the commercial ghost)	我们一直是按双轨制在推进一个 轨叫公共轨另外一个轨叫商业轨 (We have been pushing on a two-track system one track is called the public track and the other is called the commercial track)

Figure 3: Example of results with and without speller.

training sets, denoted as *N-best(1)*, *N-best(5)* and *N-best(10)*. These three training sets contain of about 20 million (20M), 55M and 84M sentences, respectively. We list the experimental configurations and results in Table 5. During inference, the best paths from the WFST-based decoder are fed into the speller to generate the final recognition result with greedy search. The experimental conclusions are the same to greedy search CTC based experiments in Sec.4.3 that N-best based data expansion method and SGDR based multi-pass training can both significantly improve the performance. Finally, the speller trained with N-best(10) training set achieves a CER of 3.41% while the performance of baseline DFSMN-CTC-sMBR system with WFST-based decoding is 4.42%(in Table 2).

4.5. Error analysis

In order to understand the role of speller, as shown in Figure 2, we plot the number of insertion, deletion and substitution errors in the recognition results of systems with and without speller. Results demonstrate that the speller can automatic correct many substitution errors made by the front-end listener no matter decoded with or without external language model. Figure 3 shows a representative example. Results show that the speller is able to utilize the sentence-level linguistic information, which is helpful to distinguish homophone in Mandarin.

5. Conclusions

In this work, we propose a transformer based spelling correction model with encode-decoder architecture to automatically correct errors made by CTC-based speech recognition system. Experimental results show that the speller is able to utilize the sentence-level linguistic information, which is helpful to automatically correct the substitution errors in the recognition results. Moreover, we propose to extend the diversity of training data using the N-best based data expansion methods, which results in more than 10% relative improvement. Finally, experimental results in a 20,000 hours Mandarin speech recognition task show that the proposed spelling correction model can achieve a CER of 3.41%, which results in 22.9% and 53.2% relative improvement compared to the baseline CTC-based systems decoded with and without language model, respectively.

6. References

- [1] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International Conference on Machine Learning*, 2014, pp. 1764–1772.
- [2] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent NN: First results," *arXiv preprint arXiv:1412.1602*, 2014.
- [3] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Sathesh, S. Sengupta, A. Coates *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.
- [4] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015, pp. 577–585.
- [5] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.
- [6] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning*, 2016, pp. 173–182.
- [7] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4945–4949.
- [8] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4835–4839.
- [9] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.
- [10] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.
- [11] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.
- [12] Y. Miao, M. Gowayyed, and F. Metze, "EESN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 167–174.
- [13] J. Li, H. Zhang, X. Cai, and B. Xu, "Towards end-to-end speech recognition for chinese mandarin using long short-term memory recurrent neural networks," in *Sixteenth annual conference of the international speech communication association*, 2015.
- [14] A. Kannan, Y. Wu, P. Nguyen, T. N. Sainath, Z. Chen, and R. Prabhavalkar, "An analysis of incorporating an external language model into a sequence-to-sequence model," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 1–5828.
- [15] S. Zhang, M. Lei, Y. Liu, and W. Li, "Investigation of modeling units for mandarin speech recognition using DFSMN-CTC-SMBR," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019.
- [16] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.
- [17] X. Liu, X. Chen, Y. Wang, M. J. Gales, and P. C. Woodland, "Two efficient lattice rescoring methods using recurrent neural network language models," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 8, pp. 1438–1449, 2016.
- [18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.
- [19] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.
- [20] R. Errattahi, A. El Hannani, and H. Ouahmane, "Automatic speech recognition errors detection and correction: A review," *Procedia Computer Science*, vol. 128, pp. 32–37, 2018.
- [21] L. Zhou, Y. Shi, J. Feng, and A. Sears, "Data mining for detecting errors in dictation speech recognition," *IEEE transactions on speech and audio processing*, vol. 13, no. 5, pp. 681–688, 2005.
- [22] A. Allauzen, "Error detection in confusion network," in *Eighth Annual Conference of the International Speech Communication Association*, 2007.
- [23] T. Pellegrini and I. Trancoso, "Error detection in broadcast news asr using markov chains," in *Language and Technology Conference*. Springer, 2009, pp. 59–69.
- [24] A. Sarma and D. D. Palmer, "Context-based speech recognition error detection and correction," in *Proceedings of HLT-NAACL 2004: Short Papers*. Association for Computational Linguistics, 2004, pp. 85–88.
- [25] Y. Bassil and P. Semaan, "ASR context-sensitive error correction based on microsoft n-gram dataset," *arXiv preprint arXiv:1203.5262*, 2012.
- [26] T. Zenkel, R. Sanabria, F. Metze, J. Niehues, M. Sperber, S. Stüker, and A. Waibel, "Comparison of decoding strategies for ctc acoustic models," *arXiv preprint arXiv:1708.04469*, 2017.
- [27] J. Guo, T. N. Sainath, and R. J. Weiss, "A spelling correction model for end-to-end speech recognition," *arXiv preprint arXiv:1902.07178*, 2019.
- [28] S. Zhang, M. Lei, Z. Yan, and L. Dai, "Deep-FSMN for large vocabulary continuous speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5869–5873.
- [29] S. Zhang, C. Liu, H. Jiang, S. Wei, L. Dai, and Y. Hu, "Feed-forward sequential memory networks: A new structure to learn long-term dependency," *arXiv preprint arXiv:1512.08301*, 2015.
- [30] H. Sak, F. de Chaumont Quiry, T. Sainath, K. Rao *et al.*, "Acoustic modelling with cd-ctc-smb LSTM RNNs," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 604–609.
- [31] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech & Language*, vol. 16, no. 1, pp. 69–88, 2002.
- [32] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "OpenNMT: Open-source toolkit for neural machine translation," in *Proc. ACL*, 2017. [Online]. Available: <https://doi.org/10.18653/v1/P17-4012>
- [33] S. Zhang and M. Lei, "Acoustic modeling with DFSMN-CTC and joint CTC-CE learning," *Proc. Interspeech 2018*, pp. 771–775, 2018.