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

Spoken Language 'Grammatical Error Correction'

Yiting Lu, Mark J.F. Gales, Yu Wang

ALTA Institute / Engineering Department, University of Cambridge, UK
{ytl28,mjfg,yw396}@cam.ac.uk

Abstract

Spoken language 'grammatical error correction' (GEC) is an important mechanism to help learners of a foreign language, here English, improve their spoken grammar. GEC is challenging for non-native spoken language due to interruptions from disfluent speech events such as repetitions and false starts and issues in strictly defining what is acceptable in spoken language. Furthermore there is little labelled data to train models. One way to mitigate the impact of speech events is to use a disfluency detection (DD) model. Removing the detected disfluencies converts the speech transcript to be closer to written language, which has significantly more labelled training data. This paper considers two types of approaches to leveraging DD models to boost spoken GEC performance. One is sequential, a separately trained DD model acts as a pre-processing module providing a more structured input to the GEC model. The second approach is to train DD and GEC models in an end-to-end fashion, simultaneously optimising both modules. Embeddings enable end-to-end models to have a richer information flow. Experimental results show that DD effectively regulates GEC input; end-to-end training works well when fine-tuned on limited labelled in-domain data; and improving DD by incorporating acoustic information helps improve spoken GEC.

Index Terms: grammatical error correction, disfluency detection

1. Introduction

The problem of automatic assessment of second language acquisition has been wildly studied in computer-assisted language learning (CALL). Among others, grammatical construction is one of the key aspects of assessing learner English, and GEC has attracted considerable interest over the past few years [1, 2, 3]. Phrase-based statistical machine translation (SMT) [4, 5], and more recently neural machine translation (NMT) models [6, 7, 8] have both achieved high performance in GEC. Previous work has mostly been focusing on correcting errors in written text. With spoken communication skills playing a big part in language acquisition, it is also important to give feedback to learners on their use of spoken grammar. Despite the fact that no strict rules are followed in free speaking, there are nonetheless phrases that a native speaker is highly unlikely to say, and feedback on these 'grammatical errors' helps learners to reflect on their spoken language.

Text-based GEC does not generalise well to speech transcripts. Spontaneous speech often comes with speech events, such as repetitions and false starts [9]. They interrupt the speech flow, complicate the grammatical structure and thus disrupt the error correction process. With little labelled speech data, direct training is not feasible. Previous work in low resource scenarios uses synthetic techniques to generate artificial data [10, 11], the quality of which largely depends on hand-crafted rules or previously trained models. In contrast, utilising a DD model as a pre-processing module mitigates the impact of disfluencies without additional complications in synthesis. By removing interrupted regions, DD regulates speech transcripts to become more text-like, and allows use of labelled text data in GEC training. Parsing based methods [12], sequence tagging models with hand-crafted features [13] as well as modified sequence-to-sequence models [14] are proved effective in tackling DD.

This paper extends previous work on modular DD pre-processing [15] to explore more options for leveraging DD models to help improve spoken GEC. The aim is to convert non-native disfluent speech into native-like fluent English. Readily available corpora allows DD training on native spoken English, and GEC training on learner written text. Under the constraint of domain mismatch, two types of approaches are considered. One is sequential, where DD and GEC are separately trained in their respective domain, and consecutively handles disfluencies and grammatical errors. Having two stand-alone modules, the cascaded structure effectively avoids interference arising from different training objectives, yet it suffers from error propagation as are most cascaded structures. The second approach is to train DD and GEC in an end-to-end fashion. While reserving a sequential structure, the end-to-end model allows gradients to back propagate through module connection and simultaneously optimising both modules. Compromising between the two objectives might yield a sub-optimal solution, yet end-to-end models surpass cascaded structures in reducing error propagation, and its flexibility in use of embedding connection also allows a richer information flow. To further apply spoken GEC on transcriptions generated through automatic speech recognition (ASR), acoustic information is incorporated into DD. Two types of attention mechanism are used to align transcripts with relevant speech segments. One uses local attention with word-level timestamps; and the other uses global pyramid attention pre-trained with end-to-end speech recognition Listen, Attend and Spell (LAS) [16]. Our contributions in this paper are: 1. we investigate different models for improved spoken GEC; confirm DD is an effective method of regulating speech transcripts; 2. we fine-tune systems on limited in-domain data, and discover that embeddings in end-to-end systems allow better adaptation to the target domain; 3. we propose to incorporate attention over acoustics into sequence tagging DD, and report positive impact on both manual and ASR transcriptions.

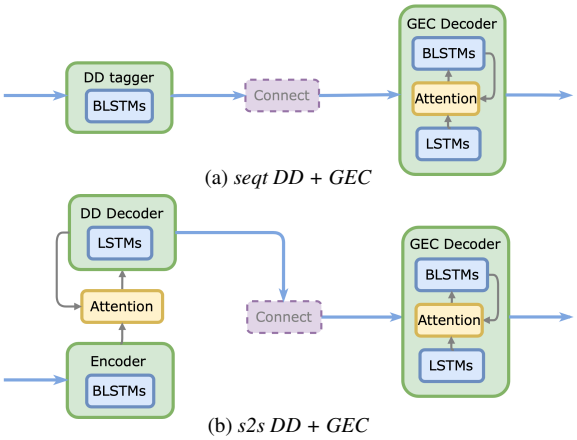
2. Network Structures

Spoken GEC converts learner speech into native text through disfluency removal followed by grammar correction. Four domains are involved in this task, namely: native speech $n^s_{1:M}$, native text $n^t_{1:Q}$, learner speech $l^s_{1:K}$ and learner text $l^t_{1:P}$. The available parallel data allows training for native speech DD $\{n^s \rightarrow n^t\}$, and text-based GEC $\{l^t \rightarrow n^t\}$. Yet spoken GEC operates between the unseen pair $\{l^s, n^t\}$ which often requires

This paper reports on research supported by Cambridge Assessment, University of Cambridge. Thanks to Cambridge English Language Assessment for support and access to the BULATS data.

http://dx.doi.org/10.21437/Interspeech.2020-1852

sequential processing $\{l^s \rightarrow l^t \rightarrow n^t\}$. Different model structures are investigated for this unseen task.



(a) *seqt DD + GEC*

(b) *s2s DD + GEC*

Figure 1: *Overview of cascade and E2E structures with two DD types. Purple blocks represent module connections.*

Baseline The baseline is a text-based GEC, i.e. a vanilla NMT model trained on the learner-native text pairs $\{l^t_{1:P} \rightarrow n^t_{1:Q}\}$. Here a standard LSTM-based attention encoder-decoder structure [17] is used. $\mathbf{d}$, $\mathbf{s}$ are the respective encoder, decoder hidden states; a is the attention weight and $\mathbf{c}$ is the context vector:

$$\mathbf{d}_{1:P} = \text{BLSTM}(l^t_{1:P}) \quad \mathbf{s}_q = \text{LSTM}(\mathbf{s}_{q-1}, n^t_q, \mathbf{c}_q)$$

$$a_{q,p} = \text{att}(\mathbf{s}_{q-1}, \mathbf{d}_p) \quad \mathbf{c}_q = \sum_{p=1}^{P} a_{q,p}\mathbf{d}_p$$

$$p(n^t_q | n^t_{<q}, l^t_{1:P}) = \text{softmax}(g(n^t_{q-1}, \mathbf{s}_{q-1}, \mathbf{c}_q))$$

Multi-style Without seeing disfluencies, the baseline GEC tends to suffer significant degradation when operating on speech transcriptions. To take into account disfluencies, the native speech-text pairs $\{n^s_{1:M} \rightarrow n^t_{1:Q}\}$ are directly blended together with the learner-native text pairs $\{l^t_{1:P} \rightarrow n^t_{1:Q}\}$ to train the multi-style model, and the model structure is kept as a vanilla NMT. The idea of multi-style training is borrowed from multilingual NMT enabling zero-shot translation [18], where the unseen task benefited from interlingua representation through diversifying the source and target domains. Although data blending allows GEC to be trained on disfluencies, the multi-style model neglects the sequential nature of $\{l^s_{1:K} \rightarrow l^t_{1:P} \rightarrow n^t_{1:Q}\}$. The stand-alone structure is unable to separate DD and GEC, and loses the interpretability to produce learner text $l^t_{1:P}$ as an intermediate output.

Cascade Unlike the implicit modeling of disfluency removal in the multi-style model, the cascade model adopts an explicit DD module trained on $\{n^s_{1:M} \rightarrow n^t_{1:Q}\}$ pairs, and connects it to a text-based GEC trained on $\{l^t_{1:P} \rightarrow n^t_{1:Q}\}$. Figure 1 shows the model structures with two different DD configurations, both using words to connect the DD and GEC modules. DD can be modelled as a simple sequence tagging (seqt) task (Figure 1a):

$$\mathbf{d}_{1:M} = \text{BLSTM}(n^s_{1:M}) \quad p(r_m | n^s_{1:M}) = f_d(\mathbf{d}_m)$$

where r_m is a binary tag indicating whether word n^s_m is disfluent. All words tagged as disfluencies will be removed from downstream processing. An alternative sequence-to-sequence (s2s) DD model is also considered (Figure 1b). It follows the conventional NMT, and the translation process simply removes disfluencies. The cascade structure allows partial in-domain training, i.e. the training and evaluation domain of text-based GEC module stays the same; whereas DD is expected to adapt to learner speech despite being trained on native speech. With the two modules being separately optimised, the training of text GEC won't be disrupted by the domain mismatch in DD, though in evaluation, extra DD errors arising from domain mismatch will propagate through and degrade text GEC.

End-to-end (E2E) In contrast with the cascade model, end-to-end training simultaneously optimises for DD and GEC. E2E training aims to mitigate error propagation by allowing gradients to flow through module connection (purple block in Figure 1) while keeping the sequential structure $\{l^s \rightarrow l^t \rightarrow n^t\}$. E2E models often require parallel data, however, the learner speech-text pairs $\{l^s \rightarrow l^t\}$ are not available for training. The comprise is to keep using native speech-text pairs $\{n^s_{1:M} \rightarrow n^t_{1:Q}\}$ for DD, while piping learner text $l^t_{1:P}$ through both DD and GEC modules to yield native text $n^t_{1:Q}$:

$$n^s_{1:M} \xrightarrow{\text{DD}} n^t_{1:Q}$$

$$l^t_{1:P} \xrightarrow{\text{DD}} \hat{l}^t_{1:P} \xrightarrow{\text{GEC}} n^t_{1:Q}$$

Applying DD on learner text $l^t_{1:P}$ should lead to an approximately unchanged sequence $\hat{l}^t_{1:P}$. It is hoped that by exposing the DD module to learner written text, it generalises better to non-native DD in evaluation. E2E can be used with both sequence tagging and seq2seq style DDs (Figure 1). When the module connection is set as words, Gumbel softmax [19] is used to allow gradient propagation through words. However, information carried through one-best word is very restrictive, and any error in prediction tends to cause large disruption to downstream GEC. To make the connection more flexible, it is possible to connect through embeddings when seq2seq style DD is adopted. In an encoder-decoder framework, the attention mechanism aligns the encoded input sequence with the current decoder state, and generates a continuous hidden vector that is analogous to an embedding. This embedding contains the complete back history of the previous context, and gives a soft representation of the current prediction. Such embedding connection allows a much richer information flow between DD and GEC.

3. Improved Disfluency Detection

Text-based DD removes disfluencies from manual transcriptions with high accuracy, and yet its performance significantly degrades when operating on ASR transcriptions. To help moderate disruption caused by ASR errors, additional acoustic information can be used. Previous work has looked at adding hand-crafted acoustic cues to help improve DD [20], here filter bank (FBK) features are used. FBK features often are thousands of frames long. To match relevant segment to each word, acoustics needs to be aligned with word sequences, and two attention mechanisms were considered for the alignment task.

Timestamps Traditional hybrid ASR produces transcriptions as well as corresponding timestamps, specifying the start and end point of each word. One simple way of extracting acoustic features is to run a local attention mechanism over each word period, masking out the rest of the sentence. Assuming input acoustic features $\mathbf{v}_{1:T}$ is decoded as $y_{1:N}$ using hybrid ASR:

$$\mathbf{h}_t = \text{BLSTM}(\mathbf{h}_{t-1}, \mathbf{v}_{1:T})$$

$$\mathbf{c}_n = \text{Att}(y_n, \mathbf{h}_{n_1:n_2})$$

where $\mathbf{h}_t$ is an acoustic-level hidden state, $[n_1, n_2]$ is the timestamp for word y_n.

LAS Listen, attend and spell [16] is an end-to-end speech recogniser framework. The encoder-decoder structure trains an attention mechanism over acoustics, which effectively offers automatic alignment between acoustic features and word tokens. The idea is to make use of the attention alignment trained with LAS to extract relevant acoustics and combine with DD.

$$\text{listen: } \mathbf{h}_t^j = \text{pBLSTM}(\mathbf{h}_{t-1}^j, [\mathbf{h}_{2t}^{j-1}, \mathbf{h}_{2t+1}^{j-1}])$$

$$\mathbf{h}_{1:T}^0 = \text{BLSTM}(\mathbf{v}_{1:T}) \quad j = 1, 2, 3$$

$$\text{attend: } \mathbf{c}_n = \text{Att}(\mathbf{s}_n, \mathbf{h}_{1:T/8}^3) \quad \mathbf{s}_n = \text{RNN}(\mathbf{s}_{n-1}, y_{n-1}, \mathbf{c}_{n-1})$$

$$\text{spell: } p(y_n|\mathbf{v}_{1:T}, y_{<n}) = f_a(\mathbf{s}_n, \mathbf{c}_n)$$

where each pyramid BLSTM (pBLSTM) layer reduces time resolution by a factor of 2, and a total of 3 layers reduces resolution by 8. The attend an spell steps follow the standard RNN decoder structure. LAS-based attention does not require explicit timestamps and has the flexibility to attend over a variable sequence length. Word error rate (WER) of LAS is often higher than hybrid ASR, therefore the context vector $\mathbf{c}_n$ used for DD is generated under teacher forcing mode with the reference transcription produced using hybrid ASR.

The extracted context vector $\mathbf{c}_n$ is then concatenated with word-level features, and further used for DD classification.

$$\mathbf{d}_n = [\text{BLSTM}(y_{1:N})_n, \mathbf{c}_n]$$

$$p(r_n|\mathbf{v}_{1:T}, y_{1:N}) = f_d(\mathbf{d}_n)$$

4. Experimental setup

4.1. Corpora and Metrics

Switchboard [21] consists of approximately 260 hours of telephone conversations of native English speakers. The Treebank-3 corpus [22] provides Switchboard transcripts as well as disfluency annotations. For DD training, the corpus is divided into the standard DD train/dev/test sets [23]. For LAS and hybrid ASR training, the standard Switchboard-300 partition is used excluding the DD dev and test sets. To extract acoustics for DD training, the Treebank-3 transcriptions are aligned with Switchboard-300 (Mississippi State transcriptions), and annotations mapped. Alignment is done at the per speaker level due to segmentation mismatch between the two corpora.

Cambridge Learner Corpus (CLC) [24] consists of written examinations of candidates at different proficiency levels with 86 different mother tongues. Grammatical errors were carefully annotated, through which reference sentences are generated and used for GEC training. Spelling mistakes, punctuation and capitalisation were removed to make the written corpus consistent with speech transcriptions.

NICT-JLE [25] is a publicly available non-native speech corpus. It provides manual transcriptions of an English oral proficiency interview involving Japanese English learners at A1-B2 levels on the CEFR scale [26]. NICT is annotated with disfluencies and grammatical errors, but the original audio recordings are not released.

BULATS [27] is a proprietary spoken corpus derived from a free speaking business English test consisting of prompted responses of up to 1 minute. The 225 learners are from 6 L1s and have a even distribution across all speaking CEFR grades. BULATS manual transcriptions are annotated with metadata, error types and corrections. Some words are annotated as unknown error, which leads to ambiguity in categorising grammatical errors and disfluencies. Various statistics of each corpus are listed in Table 1.

	SWBD	CLC	NICT	BLTS	
Average Length	8	16	7	17	
Average Disf. Length	1.8	-	2.4	1.9	
Percentage Disf.	11.1	-	13.6	9.2	3.4 (disf.) 5.8 (unk)

Table 1: *corpora statistics*

Following previous work, F_1 score is used to evaluate DD. M^2 [28] and GLEU [29] scores are commonly used metrics for GEC. M^2 requires reference edits for scoring. It is difficult to map reference edits from manual annotations to ASR outputs due to potential misalignment between manual and ASR transcripts. GLEU only requires source and target sentences to measure the n-gram edit statistics, and is therefore used here.

4.2. Models

Various models are trained using three main building blocks [1]: a sequence tagger, an encoder-decoder structure, and an LAS. A 200D word embedding is used across all models, which is initialised using GloVe [30] pretrained on Wikipedia+Gigaword5. The sequence tagger is a 2-layer 300D BLSTM followed by a binary classifier, and dropout is set at 0.5. In the encoder-decoder structure, the encoder uses a 2-layer 200D BLSTM, and the decoder is a 4-layer 200D LSTM. Decoding uses one best predictions. GEC uses bilinear attention; and DD uses monotonic [31] attention. The LAS model is implemented following [16]. Word-level targets are used and decoding is done without rescoring since the focus is to achieve a reasonable attention mechanism over acoustics. Acoustic features are 40-dimensional filter banks. The acoustic encoder consists of a 1-layer BLSTM and a 3-layer pBLSTM, both are 256 dimensional. This encoder was later used in acoustic-assisted DD without further update. The decoder uses bilinear attention, followed by a 4-layer 200D LSTM. Speaker level normalisation and spec augmentation [32] are used. The LAS has a word error rate (WER) of 33.3% on the Switchboard split of Eval2000. All models are trained using Adam optimiser [33] with a batch size of 256, and a learning rate of 0.001 with gradient clipping. Dropout is set at 0.2 if not specified.

Two ASR systems are built for Switchboard and BULATS. Both use a TDNN-F model followed by a trigram lattice generation. Switchboard is rescored with a 4-gram language model, and BULATS uses a succeeding word RNNLM [34] for rescoring. The WER is 15.6% on Switchboard and 19.5% on BULATS. Word-level timestamps are generated alongside ASR decoding; and for manual transcriptions, timestamps are obtained through force alignment.

5. Results

5.1. Base models

DD, GEC performance of various model structures are listed in Table 2. SWBD, CLC are from the respective training domain; and NICT, BULATS are used for evaluation only. Compared with the baseline, multi-style training degrades GEC on both CLC and BULATS while improving NICT by 3.4 GLEU. On NICT, higher F_1 always leads to higher GLEU regardless of model structures. Both observations suggest that disfluency re-

[1] https://github.com/EdieLu

moval largely impacts downstream GEC on NICT. As shown in Table 1, NICT has a much higher percentage disfluency compared to BULATS. This further confirms that NICT puts heavy emphasis on disfluency detection, which potentially diminishes the motivation in jointly improving DD and GEC. The following discussion is therefore focused on BULATS.

Between the two cascade models, seq2seq DD performs much worse than sequence tagging DD, and consequently degrades GEC. The nature of DD task requires only deletion, and the seq2seq structure is inherently disadvantaged due to its flexibility in substitution and insertion. However, between the two E2E models with word connection, although seq2seq DD drops F_1 by 10.6 from sequence tagging, GEC gains back 1.1 GLEU. It is mainly because sequence tagging DD cannot receive gradients propagated back from GEC (once the binary decision is made, the original word embedding is passed on to GEC, bypassing the sequence tagger), and yet errors in DD propagate downwards and degrade GEC. In comparison, seq2seq DD allows gradients to flow back, and therefore resulted in higher GLEU scores despite the disadvantage in a seq2seq structure. The deficiency in using sequence tagging DD in an E2E framework also explains why E2E training yielded a much lower GLEU score than the cascade model, provided that the same seqt-word configuration is adopted. One of the advantages of E2E training is to allow the use of embedding connection. Replacing words with embeddings in E2E training largely boosts GEC by 6.8 points, leaving a gap of 0.9 GLEU from the cascade seqt-word model despite the huge disadvantage of 17.5 F_1. This result confirms that by allowing a richer information flow, embedding connection helps compensate the disadvantage posed by seq2seq DD, and potentially enables the flexibility to fix errors made at the DD stage.

Without any in-domain training, the cascade seqt-word model performs the best. The baseline and multi-style models lack the explicit modeling of disfluencies. E2E models suffer from trade-off between DD and GEC objectives, yet the embedding connection shows potential in mitigating error propagation. Cascade structure offers strong regularisation against domain mismatch, and yet it restricts the information passed between two modules to be only words.

			DD			GEC		
Model	**DD**	**Conn**	**SWBD**	**NICT**	**BLTS**	**CLC**	**NICT**	**BLTS**
Baseline	-	-	-	-	-	**65.33**	44.58	48.95
Multi	-	-	-	-	-	62.54	47.97	47.15
Cascade	seqt	word	81.87	63.96	53.94	64.13	**52.27**	**51.36**
	s2s	word	75.51	53.54	44.63	58.21	48.58	47.70
E2E	seqt	word	79.24	61.75	48.15	50.54	49.05	42.54
	s2s	word	76.37	47.44	37.56	55.00	47.72	43.65
	s2s	emb	75.40	46.74	37.13	63.22	46.43	50.47

Table 2: *Base models in F_1 and GLEU (Manual transcriptions)*

5.2. Fine-tuning

With a small amount of annotated learner speech, it is possible to run fine-tuning with 10-fold cross-validation. For each source sentence, there are two levels of annotation, one being disfluency tags, and the other being the target grammatically correct fluent sentence. For learner speech, often it is more difficult to tag each word with its error type than simply generating a fluent target sentence. Therefore fine-tuning is carried out without using reference disfluency tags.

Table 3 shows the fine-tuning results on BULATS. The E2E model with s2s-emb structure outperforms the multi-style model by 3.9, and the cascade s2s-word model by 0.6 GLEU. Although trained end-to-end, the multi-style model is constraint by the limited amount of parallel data due to lack of explicit DD modeling. In comparison, E2E model provides a more structured end-to-end pipeline, which uses two separate attention mechanisms, each focusing on DD and GEC tasks respectively. When there is no reference disfluency tags provided, fine-tuning on the cascade model is difficult since separate optimisation is not possible without the intermediate target. The sub-optimal solution is to use the base DD module to generate hypotheses of fluent sentences, and only fine-tune GEC in the target domain. Various pre-processing tends to accumulate errors, and therefore fine-tuning on the cascade model is much less effective. On the contrary, the E2E model can be easily adapted to either scenarios by switching the DD objective on or off. E2E model further allows the use of embedding connection, which encapsulates the full back history of the DD decoder and the richer information flow helps the model to swiftly adapt to the target domain.

Model	**DD**	**Conn**	**Base**	**Fine-tune**
Multi	-	-	47.15	50.19
Cascade	seqt	word	**51.36**	53.45
E2E	s2s	emb	50.47	**54.10**

Table 3: *BULATS in GLEU (Manual transcriptions)*

5.3. Impact of acoustics

Table 4 compares DD performance on BULATS with additional acoustic information. Adding acoustic information consistently improves F_1 scores and consequently benefits GLEU on both manual and ASR transcriptions. Local attention with explicit timestamps does not perform as well as global attention trained with LAS. Timestamps tend to cut-off sharply at the word boundaries, whereas LAS attention is more flexible. Acoustic features that are particularly informative to disfluency detection often lie across word boundaries. On ASR transcriptions, DD improves spoken GEC by 1.39 and adding LAS-based attention gains another 0.45 GLEU. However, comparing across performance on manual and ASR transcriptions, there is still a huge gap of 20 GLEU caused by ASR errors.

			DD		GEC	
Model	**DD**	**Acous**	**MAN**	**ASR**	**MAN**	**ASR**
Baseline	-	-	-	-	48.95	29.82
Cascade	seqt	none	53.94	40.71	51.36	31.21
		timestamp	56.43	42.50	51.54	31.35
		LAS	57.41	43.98	51.88	31.66

Table 4: *BULATS in F_1, GLEU (Manual, ASR transcriptions)*

6. Conclusions

This paper investigates different model structures for spoken GEC, where little labelled data is available. Direct multi-style training falls short with limited data. A cascaded pipeline effectively regulates speech transcriptions, yet it still suffers from error propagation. A structured end-to-end model with embedding connection provides a richer information flow as well as the flexibility in domain adaptation. It is shown to be most effective when fine-tuned on a small amount of parallel data. Furthermore, incorporating acoustic information is shown to be useful in improving spoken GEC performance.

7. References

[1] C. Bryant, M. Felice, Ø. E. Andersen, and T. Briscoe, "The bea-2019 shared task on grammatical error correction," in *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 2019, pp. 52–75.

[2] H. T. Ng, S. M. Wu, T. Briscoe, C. Hadiwinoto, R. H. Susanto, and C. Bryant, "The conll-2014 shared task on grammatical error correction," in *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, 2014, pp. 1–14.

[3] R. Dale and A. Kilgarriff, "Helping our own: The hoo 2011 pilot shared task," in *Proceedings of the 13th European Workshop on Natural Language Generation*. Association for Computational Linguistics, 2011, pp. 242–249.

[4] R. Zens, F. J. Och, and H. Ney, "Phrase-based statistical machine translation," in *Annual Conference on Artificial Intelligence*. Springer, 2002, pp. 18–32.

[5] M. Junczys-Dowmunt and R. Grundkiewicz, "Phrase-based machine translation is state-of-the-art for automatic grammatical error correction," *arXiv preprint arXiv:1605.06353*, 2016.

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

[7] Z. Yuan and T. Briscoe, "Grammatical error correction using neural machine translation," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 380–386.

[8] A. Schmaltz, Y. Kim, A. M. Rush, and S. M. Shieber, "Adapting sequence models for sentence correction," *arXiv preprint arXiv:1707.09067*, 2017.

[9] E. E. Shriberg, "Preliminaries to a theory of speech disfluencies," Ph.D. dissertation, Citeseer, 1994.

[10] R. Grundkiewicz, M. Junczys-Dowmunt, and K. Heafield, "Neural grammatical error correction systems with unsupervised pre-training on synthetic data," in *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 2019, pp. 252–263.

[11] F. Stahlberg, "The roles of language models and hierarchical models in neural sequence-to-sequence prediction," Ph.D. dissertation, University of Cambridge, 2020.

[12] M. Ballesteros, Y. Goldberg, C. Dyer, and N. A. Smith, "Training with exploration improves a greedy stack-lstm parser," *arXiv preprint arXiv:1603.03793*, 2016.

[13] V. Zayats, M. Ostendorf, and H. Hajishirzi, "Disfluency detection using a bidirectional lstm," *arXiv preprint arXiv:1604.03209*, 2016.

[14] F. Wang, W. Chen, Z. Yang, Q. Dong, S. Xu, and B. Xu, "Semi-supervised disfluency detection," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 3529–3538.

[15] Y. Lu, M. J. Gales, K. Knill, P. Manakul, and Y. Wang, "Disfluency detection for spoken learner english." in *SLaTE*, 2019, pp. 74–78.

[16] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

[17] 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.

[18] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado *et al.*, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 339–351, 2017.

[19] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[20] V. Zayats and M. Ostendorf, "Giving attention to the unexpected: using prosody innovations in disfluency detection," *arXiv preprint arXiv:1904.04388*, 2019.

[21] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1. IEEE, 1992, pp. 517–520.

[22] A. Taylor, M. Marcus, and B. Santorini, "The penn treebank: an overview," in *Treebanks*. Springer, 2003, pp. 5–22.

[23] M. Johnson and E. Charniak, "A tag-based noisy channel model of speech repairs," in *Proceedings of the 42nd annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2004, p. 33.

[24] D. Nicholls, "The cambridge learner corpus: Error coding and analysis for lexicography and elt."

[25] E. Izumi, K. Uchimoto, and H. Isahara, "The nict jle corpus: Exploiting the language learners' speech database for research and education," *International Journal of the Computer, the Internet and Management*, vol. 12, no. 2, pp. 119–125, 2004.

[26] D. Little, "The common european framework of reference for languages: A research agenda," *Language Teaching*, vol. 44, no. 3, pp. 381–393, 2011.

[27] L. Chambers and K. Ingham, "The BULATS online speaking test," *Research Notes*, vol. 43, pp. 21–25, 2011. [Online]. Available: http://www.cambridgeenglish.org/images/23161-research-notes-43.pdf

[28] D. Dahlmeier and H. T. Ng, "Better evaluation for grammatical error correction," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2012, pp. 568–572.

[29] C. Napoles, K. Sakaguchi, M. Post, and J. Tetreault, "Ground truth for grammatical error correction metrics," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2015, pp. 588–593.

[30] R. JeffreyPennington and C. Manning, "Glove: Global vectors for word representation." Citeseer.

[31] A. Tjandra, S. Sakti, and S. Nakamura, "Local monotonic attention mechanism for end-to-end speech and language processing," *arXiv preprint arXiv:1705.08091*, 2017.

[32] 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.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[34] X. Chen, X. Liu, A. Ragni, Y. Wang, and M. J. Gales, "Future word contexts in neural network language models," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 97–103.