Cross-Sentence Grammatical Error Correction

Shamil Chollampatt, Weiqi Wang, and **Hwee Tou Ng**
Department of Computer Science, National University of Singapore
{shamil,weiqi}@u.nus.edu, nght@comp.nus.edu.sg

Abstract

Automatic grammatical error correction (GEC) research has made remarkable progress in the past decade. However, all existing approaches to GEC correct errors by considering a single sentence alone and ignoring crucial cross-sentence context. Some errors can only be corrected reliably using cross-sentence context and models can also benefit from the additional contextual information in correcting other errors. In this paper, we address this serious limitation of existing approaches and improve strong neural encoder-decoder models by appropriately modeling wider contexts. We employ an auxiliary encoder that encodes previous sentences and incorporate the encoding in the decoder via attention and gating mechanisms. Our approach results in statistically significant improvements in overall GEC performance over strong baselines across multiple test sets. Analysis of our cross-sentence GEC model on a synthetic dataset shows high performance in verb tense corrections that require cross-sentence context.

1 Introduction

Grammatical error correction (GEC) is the task of correcting errors in input text and producing well-formed output text. GEC models are essential components of writing assistance and proof-reading tools that help both native and non-native speakers. Several adaptations of sophisticated sequence-to-sequence learning models with specialized techniques have been shown to achieve impressive performance (Ge et al., 2018; Lichtarge et al., 2018; Chollampatt and Ng, 2018b; Junczys-Dowmunt et al., 2018).

All prior approaches to GEC consider one sentence at a time and ignore useful contextual information from the document in which it appears, unlike a human proofreader. Cross-sentence context is essential to identify and correct certain errors, mostly involving tense choice, use of the definite article 'the', and use of connectives. The example in Figure 1, from the NUS Corpus of Learner English or NUCLE (Dahlmeier et al., 2013), shows a learner-written sentence that seems correct in isolation, but actually involves a verb tense error in context. Sentence-level GEC systems fail to reliably correct such errors. Moreover, models may also benefit from the additional context by being able to disambiguate error corrections better.

OUT OF CONTEXT:
As a result, they are not convenient enough.
IN CONTEXT:
Electric cars have a very obvious shortage in their technique design. The electric cars invented in 1990 did not have a powerful battery. Due to the limitation of its weigh, size and the battery technology, the battery used in the electric cars at that time was limited to a range of 100 miles (Rogers,2003). ***As a result, they ~~are~~ were not convenient enough.*** Instead, Hydrogen fuel cell was brought up to substitute the electric battery.

Figure 1: A sentence from NUCLE appears correct out of context, but erroneous in context.

In this paper, we present the first approach to cross-sentence GEC[1]. We build on a state-of-the-art convolutional neural encoder-decoder model and incorporate cross-sentence context from previous sentences using an auxiliary encoder. The decoder attends to the representations generated by the auxiliary encoder via separate attention mechanisms and incorporates them via gating that controls the information that goes into the de-

[1] Our source code is publicly available at https://github.com/nusnlp/crosentgec.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 435–445
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

coder. Auxiliary encoders have also been used in other sequence generation tasks such as automatic post editing (Libovický and Helcl, 2017), multilingual machine translation (Firat et al., 2016), and document-level neural machine translation (Jean et al., 2017; Wang et al., 2017).

Our cross-sentence GEC model shows statistically significant improvements over a competitive sentence-level baseline across multiple test sets. We further improve the baseline and cross-sentence model by ensembling multiple models and rescoring. We also incorporate probabilities computed by BERT (Devlin et al., 2018) as a feature. The gains in performance with the cross-sentence component over the sentence-level baseline still remain even with the improvements. Our final cross-sentence model also outperforms state-of-the-art models trained on the same datasets on the CoNLL-2014 test set. They show notable improvements in correcting determiner and verb tense errors. Our analysis demonstrates that our cross-sentence model is able to accurately correct many cross-sentence verb tense errors when evaluated on a synthetic test set.

2 Incorporating Cross-Sentence Context

Consider a source document $\mathcal{S} = S_1, \ldots S_N$ made up of N source sentences, where S_k represents the kth source sentence. S_k comprises of $|S_k|$ tokens $s_{k,1} \ldots s_{k,|S_k|}$. In a standard sentence-level encoder-decoder model, the probability of a corrected target hypothesis $T_k = t_{k,1} \ldots t_{k,|T_k|}$, given the source sentence S_k and the set of model parameters Θ, is computed by

$$P(T_k|S_k, \Theta) = \prod_{i=1}^{|T_k|} P(t_{k,i}|T_{k,<i}, S_k, \Theta) \quad (1)$$

where $T_{k,<i}$ denotes previous target words $t_{k,1}, \ldots, t_{k,i-1}$. The above model assumes that the correction of S_k is independent of other sentences in the source document $\mathcal{S}$. This assumption may not always hold since cross-sentence context is required or helpful for correcting many errors. In our proposed cross-sentence model, the above conditional probability also depends on other sentences in the source document, denoted by $\mathcal{S}_{doc}$. Equation 1 is rewritten as

$$P(T_k|\mathcal{S}, \Theta) = \prod_{i=1}^{|T_k|} P(t_{k,i}|T_{k,<i}, S_k, \mathcal{S}_{doc}, \Theta) \quad (2)$$

We make an assumption to simplify modeling and consider only two previous source sentences as the cross-sentence context ($\mathcal{S}_{doc} = S_{k-1}, S_{k-2}$). We also do not explicitly consider previously corrected target sentences to avoid error propagation (Wang et al., 2017). We extend a sentence-level encoder-decoder baseline model to build our cross-sentence GEC model. We describe our baseline encoder-decoder model and the proposed cross-sentence model below.

2.1 Baseline Encoder-Decoder Framework

We use a deep hierarchical convolutional encoder-decoder model (Gehring et al., 2017) as our encoder-decoder framework. Ensembles of convolutional models have achieved high performance for GEC and are used in two recent state-of-the-art GEC models (Chollampatt and Ng, 2018b; Ge et al., 2018).

A source sentence S is embedded as $\mathbf{S} = \textsc{Emb}_{tok}(S) + \textsc{Emb}_{pos}(S)$ where $\mathbf{S} \in \mathbb{R}^{|S| \times d}$. $\textsc{Emb}_{tok}(\cdot)$ and $\textsc{Emb}_{pos}(\cdot)$ are token embeddings and learned position embeddings, respectively. The source embeddings $\mathbf{S}$ are then passed through a linear layer (denoted by $\textsc{Lin}$) before they are fed to the initial encoder layer as $\mathbf{H}^0 = \textsc{Lin}(\mathbf{S})$, where $\mathbf{H}^0 \in \mathbb{R}^{|S| \times h}$. The output of the lth encoder layer $\mathbf{H}^l \in \mathbb{R}^{|S| \times h}$, where $l = 1, \ldots, L$, is computed by passing the output of the previous layer $\mathbf{H}^{l-1}$ through a convolutional neural network ($\textsc{Conv}$) followed by gated linear unit (GLU) activation function (Dauphin et al., 2017), and residual connections (He et al., 2016):

$$\mathbf{H}^l = \text{GLU}(\textsc{Conv}(\mathbf{H}^{l-1})) + \mathbf{H}^{l-1} \quad (3)$$

Output of the encoder is a transformation of the final encoder layer output $\mathbf{H}^L$ to $\mathbf{E} \in \mathbb{R}^{|S| \times d}$.

The decoder network also consists of multiple layers. During the prediction of the $(n+1)$th word, the decoder has access to previously predicted n words $t_1, \ldots, t_n$ and their embeddings $\mathbf{T} \in \mathbb{R}^{n \times d}$. The embeddings are obtained in the same way as the encoder via separate embedding layers in the decoder. $\mathbf{T}$ passes through a linear layer to obtain the input to the initial decoder layer $\mathbf{G}^0 \in \mathbb{R}^{n \times h}$. The lth decoder layer computes an intermediate representation $\mathbf{Y}^l \in \mathbb{R}^{n \times h}$ by passing the outputs of the previous decoder layer $\mathbf{G}^{l-1}$ through a $\textsc{Conv}$ layer followed by a GLU activation:

$$\mathbf{Y}^l = \text{GLU}(\textsc{Conv}(\mathbf{G}^{l-1})) \quad (4)$$

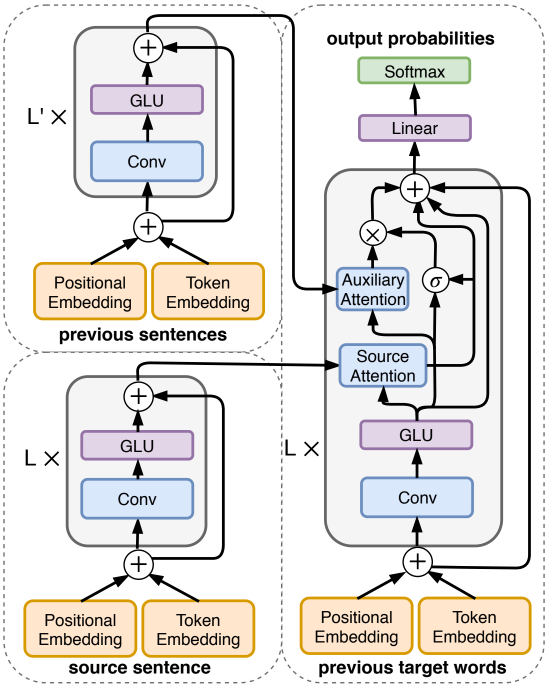

Figure 2: Our cross-sentence convolutional encoder-decoder model with auxiliary encoder and gating.

Additionally, each decoder layer has an attention mechanism that utilizes $\mathbf{Y}^l$ to compute summarized representations $\mathbf{C}^l \in \mathbb{R}^{n\times h}$ of the encoder states $\mathbf{E} \in \mathbb{R}^{|S|\times d}$ for n decoder states. Attention at layer l is computed as follows

$$\mathbf{Z}^l = \textsc{Lin}(\mathbf{Y}^l) + \mathbf{T} \tag{5}$$

$$\mathbf{X}^l = \textsc{Softmax}(\mathbf{Z}^l \cdot \mathbf{E}^\top) \cdot (\mathbf{E} + \mathbf{S}) \tag{6}$$

$$\mathbf{C}^l = \textsc{Lin}(\mathbf{X}^l) \tag{7}$$

Output of the lth decoder layer is computed as

$$\mathbf{G}^l = \mathbf{Y}^l + \mathbf{C}^l + \mathbf{G}^{l-1} \tag{8}$$

The decoder outputs a linear transformation of the final decoder layer output $\mathbf{G}^L$ to $\mathbf{D} \in \mathbb{R}^{n\times d}$. For predicting the $(n+1)$th target word, the output softmax is computed after a linear transformation of the last decoder state to the output vocabulary size. The last decoder state corresponds to the last row of $\mathbf{D}$.

2.2 Cross-Sentence GEC Model

Our proposed model (Figure 2) incorporates cross-sentence context using an auxiliary encoder that is similar in structure to our source sentence encoder (Section 2.1). For encoding the context consisting of two previous sentences, we simply concatenate these two sentences and pass it to the auxiliary encoder. Let $\hat{S}$ represent the cross-sentence context consisting of $|\hat{S}|$ tokens $\hat{s}_1, ..., \hat{s}_{|\hat{S}|}$ from the previous source sentences. They are embedded as $\hat{\mathbf{S}} \in \mathrm{R}^{|\hat{S}|\times d}$ using separate token and position embedding layers and fed into the auxiliary multi-layer encoder (consisting of L' identical layers) to obtain the auxiliary encoder output $\hat{\mathbf{E}} \in \mathrm{R}^{|\hat{S}|\times d}$.

For integrating the auxiliary encoder output during decoding, each decoder layer employs a separate attention mechanism similar to that described previously. We use another $\textsc{Lin}$ layer to compute $\hat{\mathbf{Z}}^l$ (Equation 5) and use $\hat{\mathbf{E}}$ and $\hat{\mathbf{S}}$ in place of $\mathbf{E}$ and $\mathbf{S}$, respectively, to compute summarized auxiliary encoder representations $\hat{\mathbf{C}}^l$ (Equations 6 and 7).

The summarized auxiliary encoder representation at each layer $\hat{\mathbf{C}}^l$ is added to the output of the layer along with other terms in Equation 8. All corrections do not depend equally on cross-sentence context. So, instead of adding $\hat{\mathbf{C}}^l$ directly with an equal weight as the other terms, we add a *gating* $\mathbf{\Lambda}^l \in \mathrm{R}^{n\times h}$ at each layer to control the cross-sentence information that gets passed:

$$\mathbf{G}^l = \mathbf{Y}^l + \mathbf{C}^l + \mathbf{\Lambda}^l \circ \hat{\mathbf{C}}^l + \mathbf{G}^{l-1} \tag{9}$$

where $\circ$ denotes the Hadamard product. The gate $\mathbf{\Lambda}^l$ is determined based on $\mathbf{Y}^l$ and the summarized context representations of the source sentence $\mathbf{C}^l$.

$$\mathbf{\Lambda}^l = \sigma(\textsc{Lin}(\mathbf{Y}^l) + \textsc{Lin}(\mathbf{C}^l)) \tag{10}$$

where σ represents element-wise sigmoid activation that restricts values to $[0, 1]$. $\mathbf{\Lambda}^l$ can be regarded as probabilities of retaining values in $\hat{\mathbf{C}}^l$.

We also analyze our proposed approach by comparing against two other ways for modeling cross-sentence context (Section 5.1). One way is to integrate representations from the auxiliary encoder directly in the decoder layers via the attention mechanism, without gating. Another more straightforward way of incorporating cross-sentence context is by simply passing the concatenation of the previous source sentences and the current source sentence (Tiedemann and Scherrer, 2017) to the main encoder itself, without employing an auxiliary encoder.

2.3 Other Techniques and BERT Rescoring

We further improve our models (both sentence-level baseline and cross-sentence model) with several techniques from prior work that have been shown to be useful for GEC. They include initializing the word embedding vectors with pretrained embeddings, pretraining the decoder on large English corpora using a language modeling objective (Junczys-Dowmunt et al., 2018), label smoothing (Szegedy et al., 2016), and dropping out entire word embeddings of source words during training. We found that using the technique of training with an edit-weighted objective (Junczys-Dowmunt et al., 2018) results in a higher recall but hurts the overall performance of our baseline model, and hence we do not use it in our models.

We also rescore the final candidates using feature-based rescoring (with edit operation and language model features) for both our sentence-level baseline and our cross-sentence model following Chollampatt and Ng (2018a). We investigate the effectiveness of BERT (Devlin et al., 2018) for GEC. We use *masked language model* probabilities computed by pretrained BERT model[2] as an additional feature during rescoring to further improve our sentence-level baseline and our cross-sentence GEC model. Specifically, we replace tokens in a hypothesis T with the [MASK] token, one at a time, predicting the log probability of the masked token using BERT. The final BERT feature is computed as

$$f_{\text{BERT}}(T) = \sum_{i=1}^{|T|} \log P_{\text{BERT}}(t_i|T_{-i}) \qquad (11)$$

where T_{-i} is the target sentence where the ith target word t_i is masked.

3 Experiments

3.1 Data and Evaluation

Similar to most published GEC models (Chollampatt and Ng, 2018b; Junczys-Dowmunt et al., 2018; Grundkiewicz and Junczys-Dowmunt, 2018), we rely on two datasets for training: Lang-8 Learner Corpora[3] v2.0 (Mizumoto et al., 2011) and NUCLE (Dahlmeier et al., 2013). Both datasets have document context available which make them suitable for cross-sentence GEC. We split training and development datasets based on essay boundaries. We extract development set from a subset of NUCLE. To ensure the development set has a high number of error annotations, we sort the essays in decreasing order of the ratio of corrected sentences per essay. We select 25% of essays from the top (sampling one essay from every four) until we get over 5,000 annotated sentences. The remaining essays from NUCLE are used for training. From Lang-8, we extract essays written by learners whose native language is English and contain at least two English sentences with a minimum of one annotation. We identify the language of a sentence using langid.py (Lui and Baldwin, 2012). From the extracted essays, we select annotated English sentences (with at most 80 tokens) as our source sentences and their corresponding corrections as our target sentences. Statistics of the datasets are given in Table 1. For training our cross-sentence GEC models, we select two previous English sentences for each source sentence from its essay as the cross-sentence context. We found that using two previous sentences performed better than using only one previous sentence as the context. Still, in our dataset, the first English sentence of an essay has an empty context and the second English sentence of an essay has only one previous sentence as its context.

Dataset	No. of essays	No. of sentences	No. of src tokens
Train	178,972	1,306,108	18,080,501
NUCLE	1,125	16,284	423,503
Lang-8	177,847	1,289,824	17,656,998
Dev	272	5,006	128,116

Table 1: Statistics of training and development datasets.

We evaluate on three test sets which have document-level contexts: CoNLL-2013 (Ng et al., 2013) and CoNLL-2014 (Ng et al., 2014) shared task test sets, and Cambridge Learner Corpus-First Certificate Exam or FCE test set (Yannakoudakis et al., 2011). Another recent dataset, JFLEG (Napoles et al., 2017), does not have document-level contexts available and hence we do not use it for evaluation. We use the M^{2}scorer (Dahlmeier and Ng, 2012) for evaluation and perform significance tests using one-tailed sign test with bootstrap resampling on 100 samples.

3.2 Model

We extend a convolutional encoder-decoder model following recent state-of-the-art sentence-level GEC models (Chollampatt and Ng, 2018b; Ge et al., 2018) with identical architecture and hyperparameters to build our sentence-level baseline.

[2] We use pretrained cased BERT base model from https://github.com/huggingface/pytorch-pretrained-BERT

[3] https://sites.google.com/site/naistlang8corpora

	CoNLL-2013			CoNLL-2014			FCE		
	P	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**
Baseline (avg)	54.51	15.16	35.88	65.16	27.13	50.88	52.89	26.85	44.29
CroSent (avg)	55.65	16.93	**38.17**	65.59	30.07	**53.06**	52.17	28.25	**44.61**
Baseline (ens)	55.67	15.02	36.12	66.45	27.12	51.51	53.71	26.79	44.72
CroSent (ens)	58.72	16.64	**38.99**	68.06	29.94	**54.25**	54.49	28.63	**46.15**
Baseline (ens+rs)	51.88	19.15	38.66	62.53	33.62	53.35	52.01	31.19	45.89
CroSent (ens+rs)	54.43	20.22	**40.67**	64.15	35.26	**55.12**	52.93	32.81	**47.15**
Baseline (ens+rs_{BERT})	52.77	19.01	38.93	64.08	34.21	54.55	53.47	30.91	46.65
CroSent (ens+rs_{BERT})	55.24	20.71	**41.43**	64.32	35.98	**55.57**	53.91	32.81	**47.77**

Table 2: Results of our proposed cross-sentence GEC model (CroSent). 'avg' row reports average precision (P), recall (R), and $F_{0.5}$ score of 4 independently trained single models. 'ens' denotes results of the 4-model ensemble. 'rs' denotes results of feature-based rescoring, and 'rs_{BERT}' additionally uses BERT feature for rescoring. All CroSent results are statistically significant compared to the corresponding Baseline results ($p < 0.001$).

The final system of Chollampatt and Ng (2018b) is an ensemble of three variants of this architecture initialized and trained differently. We replicate the best-performing variant using Fairseq[4] v0.5 on our training data. We incorporate the techniques mentioned in Section 2.3 such as source word dropout, pretraining word embedding, decoder pretraining, and label smoothing. We reuse the the pretrained word embeddings and vocabularies from (Chollampatt and Ng, 2018a) and the pretrained decoder from (Chollampatt and Ng, 2018b). Word embeddings had been pretrained on Wikipedia corpus consisting of 1.78 billion words. The decoder had been pretrained using 100 million sentences (1.42 billion words) from Common Crawl. To fit training of a single convolutional encoder-decoder model efficiently in a single Titan X GPU with 12 GB memory, we restrict each batch to a maximum of 6,000 source or target tokens per batch, apart from setting a maximum batch size of 96 instances. All other hyperparameters, pretrained models, and vocabularies are from (Chollampatt and Ng, 2018b), with source token dropout of 0.2 and label smoothing parameter of 0.1. We refer to this model as Baseline in our results. The best model is chosen based on the development set perplexity.

We extend Fairseq and implement an auxiliary encoder for modeling previous sentences. We use the source vocabulary itself as the vocabulary for the auxiliary encoder and initialize its embeddings with the same pretrained word embeddings used to initialize the source encoder. We use embeddings of size 500. We use three layers in the auxiliary encoder with the output size set to 1024. The cross-sentence context is encoded by the auxiliary encoder. In the case of the first sentence in an essay, an empty context consisting of only padding tokens and the end-of-sentence marker token is passed. We denote our cross-sentence GEC model as CroSent.

4 Results

We evaluate the performance of our cross-sentence GEC model, CroSent, and compare it to the sentence-level Baseline on three test sets in Table 2. Average single model results ('avg') show a notable improvement on CoNLL-2013 and CoNLL-2014 for our CroSent model. On FCE, a significant improvement of 0.32 $F_{0.5}$ is observed ($p < 0.001$) for a single model. CoNLL-2013 and FCE have only a single set of annotations for each sentence and hence, reference coverage could potentially underrate model performance on these test sets. Interestingly, the performance gap widens for an ensemble of models. CroSent achieves significant improvements of 2.87, 2.74, and 1.43, respectively, on the three test sets. Feature-based rescoring ('rs') shows further improvements. Adding BERT ('rs_{BERT}') improves our Baseline and CroSent model further. The improvements due to the integration of the cross-sentence component still remain notable and significant.

4.1 Comparison to the State of the Art

We compare our CroSent model to the state of the art. Our best result of 55.57 for CroSent (ens+rs_{BERT}) is competitive to the

[4] https://github.com/pytorch/fairseq

	F$_{0.5}$
Baseline (ens+rs$_{\text{BERT}}$)	54.55
CroSent (ens+rs$_{\text{BERT}}$)	**55.57**
$\star$NUS$_{1+2+3}$ (ens)	52.49
$\star$NUS$_{2+3}$+Baseline (ens)	52.77
$\star$NUS$_{2+3}$+CroSent (ens)	**54.87**
$\star$NUS$_{2+3}$+Baseline (ens+rs$_{\text{BERT}}$)	55.47
$\star$NUS$_{2+3}$+CroSent (ens+rs$_{\text{BERT}}$)	**57.16**
+ spell	**57.30**
Best published results (same datasets)	
NUS$_1$$_{+2+3}$ (2018b) (ens+rs+spell)	56.43
G&J (2018) (w/ spell)	56.25
JGGH (2018)	55.8
NUS$_1$ (2018a) (w/ spell)	54.79
Best published results (larger training datasets)	
Google (2018)	58.3
MSR (2018)	60.0

Table 3: Comparison to the best published results on the CoNLL-2014 test set. $\star$indicates controlled replication under identical setup as ours. All CroSent results are statistically significant compared to the corresponding baselines ($p < 0.001$).

best published models trained using the same training datasets: NUS$_{1+2+3}$ (Chollampatt and Ng, 2018b), G&J (Grundkiewicz and Junczys-Dowmunt, 2018), JGGH (Junczys-Dowmunt et al., 2018), and NUS$_1$ (Chollampatt and Ng, 2018a). NUS$_{1+2+3}$ is a 12-model ensemble of three sets of models (4 each), with one set reused from NUS$_1$. We re-implement this ensemble model ($\star$NUS$_{1+2+3}$) using identical preprocessing and data used by our Baseline, without rescoring and spell checking. Our replicated results reach 52.49. We replace the weakest set of 4 models in NUS$_{1+2+3}$, i.e., NUS$_1$, in the ensemble with our 4 Baseline models instead. We reach a result of 52.77. We then use our CroSent models in place of Baseline models and observe notable improvements of 2.1 F$_{0.5}$ score. When we use BERT feature and rescore, our improved baseline $\star$NUS$_{2+3}$+Baseline (ens+rs$_{\text{BERT}}$) achieves 55.47 F$_{0.5}$ score, and using CroSent models instead of our Baseline models achieves a result of 57.16 (+1.69 F$_{0.5}$). This result is better than that of all prior competing models trained on the same datasets[5], of which three systems use a spell checker from (Chollampatt and Ng, 2018a). When we add this spell checker in a similar way, our final result reaches 57.30 F$_{0.5}$.

With much larger high-quality annotated training datasets, a better result of 60.0 is achieved by MSR (Ge et al., 2018). Recently, Google (Lichtarge et al., 2018) published a higher result of 58.3 by an ensemble of big Transformers (Vaswani et al., 2017) additionally trained on Wikipedia edits (4.1 billion words). Such large-scale training is very likely to further improve our model as well. We leave it to future work to explore large-scale cross-sentence GEC.

	Dev	**CoNLL-2013**		
	F$_{0.5}$	**P**	**R**	**F$_{0.5}$**
Baseline	33.21	54.51	15.16	35.88
concat	33.41	55.14	15.28	36.23
aux (no gate)	32.99	55.10	14.83	35.69
aux (+gate)	35.68	55.65	16.93	38.17

Table 4: Average single model results comparing different strategies to model cross-sentence context. 'aux (+gate)' is used in our CroSent model.

5 Analysis

We analyze our model choices on our development data (5,006 sentences from NUCLE) and held-out CoNLL-2013 test set. We have not used CoNLL-2013 test set directly during training and hence, it can be used to evaluate the generalization performance of our models. We also analyze model outputs on the CoNLL-2013 test set.

5.1 Modeling Cross-Sentence Context

We investigate different mechanisms of integrating cross-sentence context. Table 4 shows the average single model results of our sentence-level Baseline compared to two different strategies of integrating cross-sentence context. 'concat' refers to simply prepending the previous source sentences to the current source sentence. The context and the current source sentence is separated by a special token (`<CONCAT>`). This model does not have an auxiliary encoder. 'aux (no gate)' uses an auxiliary encoder similar to our CroSent model except for gating. 'aux (+gate)' is our CroSent model (Section 2.2) which employs the auxiliary encoder with the gating mechanism. The first two variants perform comparably to our sentence-level Baseline and shows no notable gains from using cross-sentence context. When

[5] After the submission of this paper, improved results on the CoNLL-2014 benchmark have been published (Zhao et al., 2019; Lichtarge et al., 2019; Stahlberg et al., 2019).

I agree that RFID technology shall not be made available to the public for easy abuse and distorted usage. First, our privacy is at threat. Though tracking devices such as the applications in our smartphones these days can add fun and entertainment to our fast-living paced livings, this can be a double-edged sword.**We revealed our locations, for our friends to catch up with us, however we can never know who is watching us out there secretly.**

BASELINE:	We **revealed** our locations, for our friends to catch up with us, ...
CROSENT:	We **reveal** our locations, for our friends to catch up with us, however ...
REF:	We **reveal** our locations, for our friends to catch up with us, ...

Figure 3: Output showing a cross-sentence correction.

the gating mechanism is added, results improve substantially. Using the gating mechanism is crucial in our CROSENT model, as it has the ability to selectively pass information through. This shows that properly modeling cross-sentence context is essential to improve overall performance.

5.2 Ability to Correct Cross-Sentence Errors

We show an example from the CoNLL-2013 test set which involves a verb tense error that requires cross-sentence context for correction. The original sentence (with its context) and the corrections made by our BASELINE and CROSENT ensemble models are shown in Figure 3. The verb 'revealed' is to be corrected to its present tense form 'reveal' according to the annotated reference (REF). Our CROSENT model corrects this verb tense error accurately. However, our sentence-level BASELINE is unable to make this correction as it only has access to the sentence-level context.

To investigate if cross-sentence context is adequately captured by our model, we adopt a larger scale controlled evaluation. To do this, 795 well-formed sentences are obtained from multiple documents in simple English Wikipedia along with their previous sentences. We create a synthetic dataset of verb tense errors, by corrupting all verbs in these 795 sentences and replacing them by their present tense form[6], producing 1,090 synthetic verb tense errors. These errors require cross-sentence context for correction[7]. An example is shown below:

	P / R / F$_{0.5}$	n_c / n_p
BASELINE	22.86 / 8.35 / 16.96	91 / 398
CROSENT	44.60 / 20.83 / 36.31	227 / 509

Table 5: Performance on contextual verb tense errors on a synthetic test set. n_c denotes the no. of correct changes and n_p denotes the no. of proposed changes.

CONTEXT:
He got a flat tyre, and had to drive slowly back to the pits. He finished the race in fifth place.
ORIGINAL:
His championship lead **was** reduced .
CORRUPTED:
His championship lead **is** reduced.

We analyze the performance of our cross-sentence model on this dataset by passing the corrupted sentences and their contexts as input. We evaluate the ability of our model to correct the verb tense errors and recover the original sentences. The result of our BASELINE and CROSENT ensemble models on this dataset is shown in Table 5. In this dataset, we find a sharp increase in both precision and recall for our CROSENT model showing their ability to capture cross-sentence context. While the BASELINE accurately corrects 91 errors, our CROSENT model accurately corrects 227 errors. The number of proposed corrections (n_p) is also significantly higher for CROSENT. The results indicate that our cross-sentence model can identify and correct a significant number of cross-sentence errors.

5.3 Overall Performance across Error Types

We evaluate the overall performance on common error types (Figure 4) on the CoNLL-2013 dataset using ERRANT (Bryant et al., 2017). They include determiner (DET), noun number (NOUN:NUM), preposition (PREP), verb tense (VERB:TENSE), and errors that are not categorized (OTHERS). We find that the largest margins of improvements are observed on verb tense errors and determiner errors. This aligns with our expectation of cross-sentence models.

[6] Adapted from https://github.com/bendichter/tenseflow

[7] We keep the previous sentences in their actual form to analyze the model's ability in a controlled way. However, in reality, previous sentences may also contain errors.

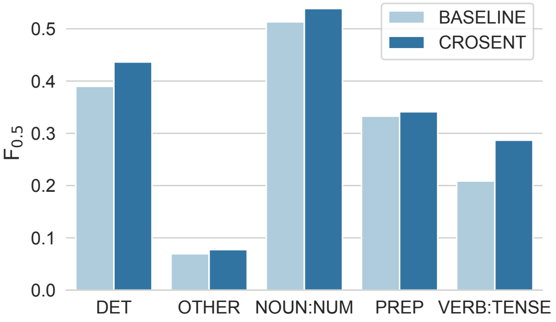

Figure 4: Performance on common error types on the CoNLL-2013 test set.

POS	Avg. KL Div.
NOUN	0.55
VERB	0.51
ADJ	0.50
DET	0.50
PROPN	0.47

Table 6: Top 5 part-of-speech (POS) tags based on average KL-divergence between auxiliary attention and uniform distribution.

5.4 Attention on Auxiliary Context

We also analyze the output words that produce attention distributions on the auxiliary context that deviate the most from a uniform distribution, indicating their dependence on cross-sentence context. For each output word, we find the KL divergence between its auxiliary attention distribution and a uniform distribution on words in the auxiliary context. We average the KL divergences for all instances of this output word in the CoNLL-2013 test set. We then identify the top words for our CROSENT ensemble model. The top words mostly include nouns (such as *criminal*, *chip*, *economy*, and *individual*). Manual analysis shows that many of these nouns appear in consecutive sentences, which results in higher attention placed on the same words on the source side.

We conduct a more coarse-grained analysis to find the top 5 part-of-speech tags of output words (based on Universal tags) that produce the highest average KL divergences (Table 6). Nouns and proper nouns (PROPN) attending to the same word in previous source sentences cause them to rank higher. Verbs and determiners (DET) are also among the top tags since they often require cross-sentence context for disambiguation. According to ERRANT (Section 5.3), compared to our BASELINE ensemble, CROSENT ensemble achieves notable improvements in performance on verb tense errors (+7.8% $F_{0.5}$) and on determiner errors (+4.7%).

6 Related Work

Sophisticated sequence-to-sequence architectures (Gehring et al., 2017; Vaswani et al., 2017) have contributed to the progress of GEC research. Employing diverse ensembles (Chollampatt and Ng, 2018b), rescoring (Chollampatt and Ng, 2018a), iterative decoding strategies (Ge et al., 2018; Lichtarge et al., 2018), synthetic (Xie et al., 2018) and semi-supervised corpora (Lichtarge et al., 2018), and other task-specific techniques (Junczys-Dowmunt et al., 2018) has achieved impressive results on this task. However, all prior work ignores document-wide context for GEC, and uses sentence-level models. For spell checking, Flor and Futagi (2012) used document-level context to check if a candidate correction for a misspelled word had been used earlier in the document. Zheng et al. (2018) proposed splitting run-on sentences into separate sentences. However, they did not use cross-sentence context.

On the other hand, there are a number of studies recently on integrating cross-sentence context for neural machine translation (NMT). There are three major approaches for document-level NMT: (1) translating an extended source (context concatenated with the source) to a single or extended target (Tiedemann and Scherrer, 2017; Bawden et al., 2018); (2) using an additional encoder to capture document-wide context (Jean et al., 2017; Wang et al., 2017; Bawden et al., 2018; Voita et al., 2018; Miculicich et al., 2018; Zhang et al., 2018); and (3) using discrete (Kuang et al., 2018) or continuous cache (Tu et al., 2018; Maruf and Haffari, 2018) mechanisms during translation for storing and retrieving document-level information. We investigated the first two approaches in this paper as we believe that most of the ambiguities can be resolved by considering a few previous sentences. Since GEC is a monolingual rewriting task, most of the disambiguating information is in the source sentence itself, unlike bilingual NMT. All approaches for document-level NMT extended recurrent models or Transformer models. There is no prior work that extends convolutional sequence-to-sequence models for document-level NMT.

The aim of this paper is to demonstrate the necessity of modeling cross-sentence context for GEC. It is beyond the scope of this paper to comprehensively evaluate all approaches to exploit document-level context, and we leave it to future work to evaluate more sophisticated models such as memory networks to capture entire document-level context or incorporate external knowledge sources.

7 Conclusion

We present the first approach to cross-sentence GEC, building on a convolutional encoder-decoder architecture. Our cross-sentence models show significant gains over strong sentence-level baselines. On the CoNLL-2014 benchmark test set, when using larger ensembles and integrating BERT during rescoring, our final cross-sentence model achieves 57.30 $F_{0.5}$ score which is higher than all prior published $F_{0.5}$ scores at the time of paper submission, when trained using the same datasets. We also demonstrate the ability of our models to exploit wider contexts adequately and correct errors on a synthetic test set of cross-sentence verb tense errors.

References

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Shamil Chollampatt and Hwee Tou Ng. 2018a. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

Shamil Chollampatt and Hwee Tou Ng. 2018b. Neural quality estimation of grammatical error correction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Daniel Dahlmeier and Hwee Tou Ng. 2012. 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*.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner English: The NUS corpus of learner English. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Michael Flor and Yoko Futagi. 2012. On using context for automatic correction of non-word misspellings in student essays. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*.

Tao Ge, Furu Wei, and Ming Zhou. 2018. Reaching human-level performance in automatic grammatical error correction: An empirical study. *arXiv preprint arXiv:1807.01270*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2018. Near human-level performance in grammatical error correction with hybrid machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Short Papers)*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does neural machine translation benefit from larger context? *arXiv preprint arXiv:1704.05135*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Shaohui Kuang, Deyi Xiong, Weihua Luo, and Guodong Zhou. 2018. Modeling coherence for neural machine translation with dynamic and topic caches. In *Proceedings of the 27th International Conference on Computational Linguistics*.

Jindřich Libovický and Jindřich Helcl. 2017. Attention strategies for multi-source sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers)*.

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jared Lichtarge, Christopher Alberti, Shankar Kumar, Noam Shazeer, and Niki Parmar. 2018. Weakly supervised grammatical error correction using iterative decoding. *arXiv preprint arXiv:1811.01710*.

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*.

Sameen Maruf and Gholamreza Haffari. 2018. Document context neural machine translation with memory networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. Mining revision log of language learning SNS for automated Japanese error correction of second language learners. In *Proceedings of the Fifth International Joint Conference on Natural Language Processing*.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: A fluency corpus and benchmark for grammatical error correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (Short Papers)*.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the 18th Conference on Computational Natural Language Learning: Shared Task*.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the 17th Conference on Computational Natural Language Learning: Shared Task*.

Felix Stahlberg, Christopher Bryant, and Bill Byrne. 2019. Neural grammatical error correction with finite state transducers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*.

Zhaopeng Tu, Yang Liu, Shuming Shi, and Tong Zhang. 2018. Learning to remember translation history with a continuous cache. *Transactions of the Association for Computational Linguistics*, 6:407–420.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. Exploiting cross-sentence context for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. 2018. Noising and denoising natural language: Diverse backtranslation for grammar correction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, and Feifei Zhai. 2018. Improving the Transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Junchao Zheng, Courtney Napoles, Joel Tetreault, and Kostiantyn Omelianchuk. 2018. How do you correct run-on sentences its not as easy as it seems. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*.