Free Writing - Grammatical Error Correction System: Sequence Tagging

Shashank[1], Shetty Shreyas Udaya[2], Sumukha N Shilge[3], Mohammed Yasir [4]
Mrs. Sreevidya B S[5]

[12345]*Department of Information Science and Engineering, Dayananda Sagar College of Engineering*

---***---

Abstract - *This paper presents a grammatical error correction (GEC) system that provides suggestions to users to make incorrect sentences to correct. Sequence tagging is a core Information extraction task in which words (or phrases) are classified using a predefined label set. The model is pre-trained on synthetically generated grammatical errors and trained on National University of Singapore Corpus of Learner English (NUCLE) (Dahlmeier et al., 2013), Lang-8 Corpus of Learner English (Lang-8) (Tajiri et al., 2012), FCE dataset (Yannakoudakis et al., 2011), the publicly available part of the Cambridge Learner Corpus (Nicholls, 2003) and Write & Improve + LOCNESS Corpus (Bryant et al., 2019). Evaluating on CoNLL2014 test set (Ng et al., 2014) evaluated by official M2 scorer (Dahlmeier and Ng, 2012), and on BEA-2019 dev and test sets evaluated by ERRANT.*

Key Words: NLP, Sequence tagging, transformers, seq2seq

1. INTRODUCTION

A neural machine translation (NMT) -based approach has emerged as the recommended method for grammatical error correction (GEC) tasks. In this formulation, incorrect statements correspond to the source language, and error-free statements correspond to the target language. Recently, Transformer-based (Vaswani et al., 2017) inter-sequence (seq2seq) models achieved cutting-edge performance in standard GEC benchmarks (Bryant et al., 2019).

Currently, the focus of research is shifting to the generation of synthetic data for pre-training Transformer NMT-based GEC systems. NMT-based GEC systems have some problems and are inconvenient to use in real-world deployment: (i) slow inference speed, (ii) requires a large amount of training data, and (iii) interpretability and explainability; NMT-based GEC system requires the additional function to explain the correctness of sentence, e.g., grammatical error type classification.

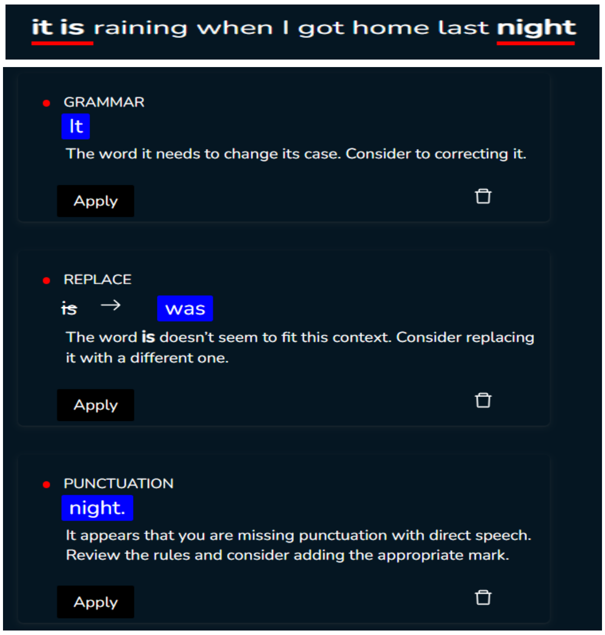
First, instead of generating a complete correct sentence from the incorrect sentence. we output edits such as copy, appends, deletes, replacements, and case-changes which generalize the task to a small size of vocabulary, unlike the NTM-based GEC system which needs a large vocabulary to generate a complete sentence. Suppose in GEC we have an input sentence: "i have dinner yesterday". Existing seq2seq learning approaches would need to output the four tokens I had dinner yesterday from a word vocabulary whereas we would predict the edits {Capitalize token 1, Replace (had) to token 2, Copy token 3, Copy token 4}

Second, we try to construct the correct sentence from those tags predicted. Each token will have the prediction token which will then apply to the tokens to form the correct sentence. From the above example "i will change to capital case I" and have will change to had.

Third, it improves the inference power of the parallel model by repeatedly feeding the output of the model itself for further improvement.

2.The GEC System

In this section, we present Free-Writing, a web-based system where users can write their essays and get suggestions to correct them. The user can ignore or apply suggestions to sentences to correct. The correction process is divided into four steps. First, we use BertTokenizer to tokenize input sentences. Second, the tokenized sentences are fed into the Bert model for inference. Third, predicted tags are converted to suggestion tokens . Finally, to give easy to-read feedback, we convert the result into an informative visual expression instead of prediction tokens.


it is raining when I got home last night

The word it needs to change its case. Consider to correcting it.

The word **is** doesn't seem to fit this context. Consider replacing it with a different one.

It appears that you are missing punctuation with direct speech. Review the rules and consider adding the appropriate mark.

Fig.1 Free Writing Web Interface for suggestion

2.1 Model Implementation

Our GEC sequence tagging model is an encoder made of pretrained BERT-like transformers stacked with two linear layers with SoftMax layers on the highest. We have used cased pretrained transformers in their Base configurations from the HuggingFace framework.

As opposition directional models, which read the text input sequentially (left-to-right or right-to-left), the Transformer encoder reads the complete sequence of words without delay. Therefore, it's considered bidirectional, though it might be more accurate to mention that it's non-directional. This characteristic allows the model to be told the context of a word supported all of its surroundings (left and right of the word).

3. Experiments

3.1 Dataset

For pre-training data, we use 2M parallel sentences with synthetically generated grammatical errors. For training data, we use following datasets National University of Singapore Corpus of Learner English (NUCLE) (Dahlmeier et al., 2013), Lang-8 Corpus of Learner English (Lang-8) (Tajiri et al., 2012), FCE dataset6 (Yannakoudakis et al., 2011), the publicly available part of the Cambridge Learner Corpus (Nicholls, 2003) and Write & Improve + LOCNESS Corpus (Bryant et al., 2019) . For Evaluation data we report results on CoNLL2014 test set (Ng et al., 2014) evaluated by official M2 scorer (Dahlmeier and Ng, 2012), and on BEA-2019 dev and test sets evaluated by ERRANT.

Dataset	#sentences
Synthetic	2,000,000
Lang-8	947,344
NUCLE	56,958
FCE	34,490
W&I+LOCNESS	34,304

Table 1: Training datasets and the number of sentences in training data

3.2 Preprocessing

To approach the task as a sequence tagging problem, we need to convert pairs of sentence into sequence and tags. Our sequence-tagging approach relies on custom transformation tags, we define these tags on tokens, which in NLP are units of text (usually words, sometimes punctuation marks, depending on how tokenization is done). For example, if a token is tagged with $DELETE, that means it should be removed from the sentence to make the text more correct. The tag $APPEND_{.} means that a period should be appended to this token to make the text more correct.

Source sentence: it is raining when I got home last night

Target sentence: It was raining when I got home last night.

it - $TRANSFORM_CASE_CAPITAL

is - $REPLACE_was

raining -$KEEP

when - $KEEP

I - $KEEP

got - $KEEP

home - $KEEP

last - $KEEP

night - $APPEND_{.}

3.3 Hyperparameters and Training Details

In our experiments, we used an early stopping mechanism and a hard and fast number of epochs n_epoch: 20 patience: 3, the matter with this approach is sensitivity to random seeds, model initialization, data order, etc. The longer you train, the upper recall you get, except for the value of precision, so it is vital to prevent training at the proper time. For reproducibility reasons, we are providing further the precise number of epochs for every model and every stage. The source sentence length is 128 and that we are using batch size 32 for training and prediction.

4. Conclusions

We have presented a GEC system that offers suggestions for erroneous sentences and shows that it's faster, simpler, and more efficient. The GEC system may be developed by employing a sequence tagging approach, an encoder from a pre-trained Transformer, and custom transformations. we've got a pre-trained model with synthetic dataset and fine-tuned with gec standard dataset.

Our GEC model achieves a $F_{0.5}$ of 65.3 / 66.5 on CoNLL-2014 (test) and $F_{0.5}$ of 72.4 / 73.6 on BEA-2019 (test). Achieve cutting-edge results for GEC tasks with inference speeds up to 10x faster than transformer-based seq2seq systems.

REFERENCES

[1] Cool English: A Grammatical Error Correction System Based on Large Learner Corpora https://www.aclweb.org/anthology/C18-2018.pdf

[2] Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer https://arxiv.org/pdf/1910.10683.pdf

[3] Attention Is All You Need https://arxiv.org/pdf/1706.03762.pdf

[4] How Good Are Grammatical Error Correction Systems? https://www.aclweb.org/anthology/2021.eacl-main.231.pdf

[5] Stronger Baselines for Grammatical Error Correction Using a Pretrained Encoder-Decoder Model https://www.aclweb.org/anthology/2020.aacl-main.83.pdf

[6] C. Park, Y. Yang, C. Lee, and H. Lim, "Comparison of the evaluation metrics for neural grammatical error correction with overcorrection," IEEE Access, vol. 8, no. 8, pp. 106264–106272, 2020.View at: Publisher Site | Google Scholar

[7] W. Yinxia and L. Yang, "A systemic functional grammar study on the embedding of prominent tendency features in evaluative "V de C" clauses," Foreign Language, vol. 43, no. 1, pp. 23–33, 2020.View at: Google Scholar

[8] J.-H. Lee, M. Kim, and H.-C. Kwon, "Deep learning-based context-sensitive spelling typing error correction," IEEE Access, vol. 8, no. 8, pp. 152565–152578, 2020.View at: Publisher Site | Google Scholar

[9] Y. Jing, "Construction and analysis of syntax error correction algorithm model based on deep learning technology," Information Technology, vol. 44, no. 9, pp. 151–155+160, 2020.View at: Google Scholar

[10] H. H. Pajouh, R. Javidan, R. Khayami, A. Dehghantanha, and K.-K. R. Choo, "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks," IEEE Transactions on Emerging Topics in Computing, vol. 7, no. 2, pp. 314–323, 2019.View at: Publisher Site | Google Scholar