Grammatical Error Correction using Neural Network Language

Prof. Uday Nayak, Chris Fernandes, Nelton Fernandes, Sonialla Lobo
Department of Information Technology
Don Bosco Institute of Technology, Kurla
Mumbai University, Maharashtra, India
chrisfdes765@gmail.com
neltonfernandes@gmail.com
soniallalobo@gmail.com

***Abstract*— Grammatical error correction (GEC) is the task of automatically correcting grammatical errors in written text. As sentences may contain multiple errors of different types, a practical error correction system should be able to detect and correct all errors. This paper describes a method for using a Language Model (LM) for Grammar Error Correction (GEC) that does not require annotated data. Specifically, Bidirectional Encoder Representations from Transformers (B.E.R.T) is being used as a Language Model.**

Keywords—Grammar Error Correction (GEC), Language Model (LM), Bidirectional Encoder Representations from Transformers (B.E.R.T), Natural Language Processing (NLP), Deep Learning (DL)

I. INTRODUCTION

Grammatical errors are of the many differing types, including articles or determiners, prepositions, noun form, verb form, subject-verb agreement, pronouns, word choice, syntax, punctuation, capitalization, etc. Of all the error types, determiners and prepositions are among the foremost frequent errors made by learners of English.[1]. the main focus of this project is to correct errors in spellings, determiners, prepositions & action verbs using B.E.R.T as a language representation model. Bidirectional Encoder Representations from Transformers (B.E.R.T) is meant to pretrain deep bidirectional representations from unlabelled text by jointly conditioning on both left and right context altogether layers. As a result, the pre-trained B.E.R.T model are often finetuned with only one additional output layer to make state-of-the-art models for a good range of tasks, like question answering and language inference, without substantial task specific architecture modifications.

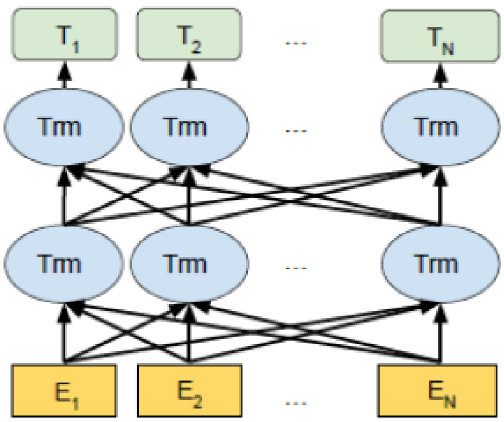
We arrange to use B.E.R.T for two tasks
• Grammar Error Detection (GED)
• Grammar Error Correction (GEC)

Specifically, for task associated with GEC, we arrange to hump without the utilization of any annotated training, and just depend upon the language knowledge captured by the B.E.R.T Masked Language Model (MLM).

II. BACKGROUND

A. B.E.R.T

One of the most important challenges in linguistic communication processing (NLP) is that the shortage of coaching data. NLP could be a diversified field with many distinct tasks, most task specific datasets contain only some thousand or some hundred thousand human labelled training examples. to assist close this gap in data, researchers have developed a range of techniques for training general purpose language representation models using large amounts of unannotated text on the online (known as pre training). The pre trained model can then be fine-tuned on small data NLP tasks like question answering and sentiment analysis, leading to substantial accuracy improvements compared to training on these datasets from scratch. In February 2018, Google open



sourced a replacement technique for NLP pre training called B.E.R.T. B.E.R.T alleviates the previously mentioned unidirectionality constraint by employing a “masked language model” (MLM) pre-training objective, inspired by the Cloze task (Taylor, 1953). The masked language model randomly masks a number of the tokens from the input, and also the objective is to predict the first vocabulary id of the masked word based only on its context. Unlike left-to right language model pre-training, the MLM objective enables the representation to fuse the left and also the right context, which allows us to pretrain a deep bidirectional Transformer.

B. PyTorch-Transformers

PyTorch-Transformers is a library of pre-trained models for Natural Language Processing (NLP). The library at present contains PyTorch executions, pre-trained model weights, use contents and transformation utilities for the accompanying models. We will utilize this for our interface to BERT and its undertakings.

III. RELATED WORK

A. Rule Based Approach

This approach mainly focuses on specifying hand-coded grammar rules, which the sentences must follow. It also involved simple pattern matching.

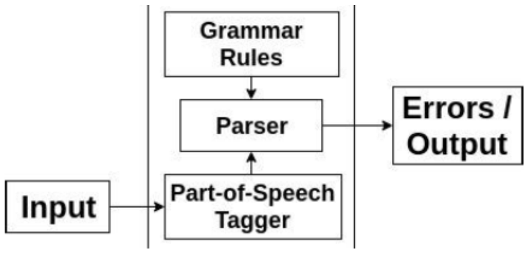


Fig. 2. Traditional Rule Based System Components

Syntactic Analysis was also incorporated with Rule Based System. Context Free Grammars are also used to specify grammar rules, and the parser is used to check the PoS tagged text according to the grammar rules defined. [8]. Although rule-based systems are easy to implement, they are unable to detect more complex errors in writings, and also the model does not generalize well, because it becomes impossible to define rules for all combination of errors.

B. Classification Based Approach

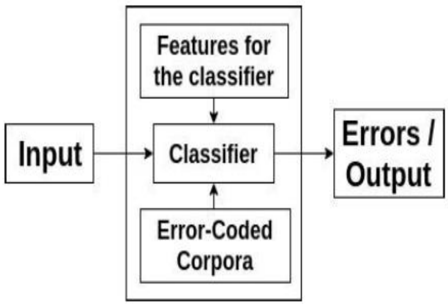


Fig. 3. Components of a Classifier based System

The availability of large error-coded corpus enabled the researchers to use more data-driven approaches for GEC. Machine Learning Algorithms were used to build classifiers for correcting specific error types. Maximum entropy model was used to determine the best possible word/replacement with respect to the prior data. [9]

In this classifier-based approach, the possible candidates i.e., words/replacements are treated as class labels, and the surrounding n-grams, PoS tags, grammatical relations are used as features. These classifiers were used to detect article errors and achieved an accuracy of 88%. [9]. Also, because the features for the classifier are dependent upon the error type, a classifier can detect only a single type of error. And it assumes that the rest of the sentence is error-free and the current error is independent, which is usually not the case.

The commonly used approach to overcome this is to build multiple classifiers, each correcting one type of error and this collection of classifiers is then cascaded into a pipeline [10]. But this approach does not work well in case of dependent errors.

IV. APPROACH

To do the task of Grammar Error Correction, we use a pre-trained B.E.R.T model, which is then fine-tuned with the Corpus of Linguistic Acceptability (CoLA) dataset for single sentence classification. It's a group of sentences labelled as grammatically correct or incorrect. We use this fine-tuned B.E.R.T to grant a model that may classify whether a sentence has grammatical error or not. this could give us the Grammar Error Detection (GED) model.

We'd then use the Masked Language Model (MLM) of B.E.R.T to return up with alternate sentences and use the GED to come back up with correction suggestions. The high-level approach would be

- Tokenize the sentence utilizing Spacy
- Check for spelling mistakes utilizing Hunspell
- For all relational word, determiners and activity action words, make a bunch of likely sentences
- Create a bunch of sentences with each word "covered", erased or an extra determiner, relational word or aide action word added
- Used BERT Masked Language Model to decide potential ideas for covers
- Use the GED model to choose

V. GRAMMAR ERROR DETECTION

There are two steps in B.E.R.T framework: pre-training and fine-tuning. During pre-training, the model is trained on unlabelled data over different pre-training tasks. For finetuning, the B.E.R.T model is first initialized with the pre-trained parameters, and all the parameters are fine-tuned using labelled data from the downstream tasks.

A. Pre-trained model

A. Pre-trained model We use the B.E.R.T-base-uncased because the pre trained model. It consists of 12-layer, 768-hidden, 12-heads, 110M parameters and is trained on lower-cased English text. We also experimented with B.E.R.T-large-uncased, which consists of 24- layer, 1024-hidden, 16-heads, 340M parameters which is trained on lower-cased English text. However, for our dataset, we failed to find any significant difference in performance.

B. Fine tuning

For fine-tuning we've used CoLA dataset for single sentence classification. The driving principle behind this approach is that the concept of The Poverty of the Stimulus. The Poverty of the Stimulus argument holds that purely data-driven learning isn't powerful enough to clarify the richness and uniformity of human grammars, particularly with data of such quality as children are exposed to. This argument is usually wielded in support of the speculation of a robust Universal Grammar, which claims that every one humans share an innately given set of language universals, which domain-general learning procedures don't seem to be enough to accumulate language (Chomsky, 1965)

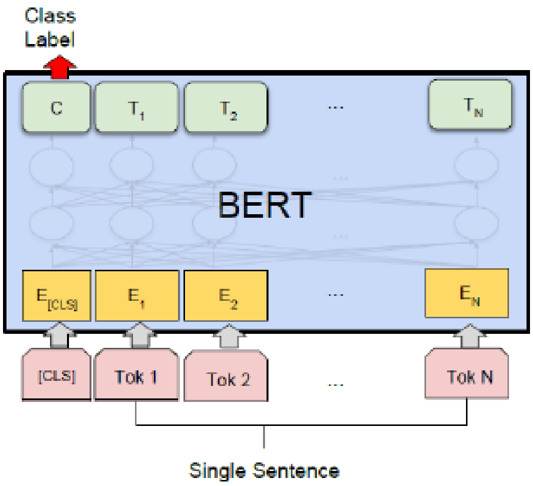


Fig. 4. B.E.R.T single sentence classification task

B.E.R.T single sentence classification task Because the pre trained B.E.R.T layers already encode lots of knowledge about the language, training the classifier is comparatively inexpensive. instead of training every layer in a very large model from scratch, it's as if we've got already trained the underside layers 95% of where they have to be, and only actually need to coach the highest layer, with a small amount of tweaking happening within the lower levels to accommodate our task. We use the subsequent hyperparameters for fine-tuning • Batch size of 32 • Learning rate (Adam): 2e-5 • Number of epochs: 4 Form the Pytorch-transformers, we use the B.E.R.TForSequenceClassification API. this can be a B.E.R.T model transformer with a sequence classification/regression head on top (a linear layer on top of the pooled output

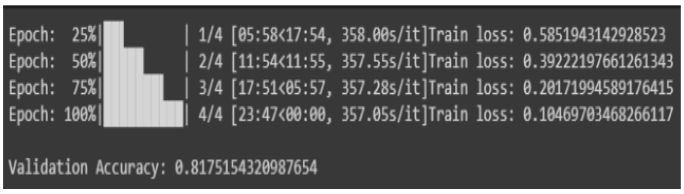


Figure 5. Training loss & validation accuracy of B.E.R.T

VI. GRAMMAR ERROR CORRECTION

We would then use the Masked Language Model (MLM) of B.E.R.T to return up with alternate sentences and use the GED to come back up with correction suggestions. The high-level approach would be:

• Tokenize the sentence using Spacy
• Check for spelling errors using Hunspell
• For all preposition, determiners & helper verbs, create a group of probable sentences
• Create a group of sentences with each word "masked", deleted or an extra determiner, preposition or helper verb added
• Used B.E.R.T Masked Language Model to work out possible suggestions for masks
• Use the GED model to pick appropriate solutions

VII. RESULTS

We used reference sentences from various papers to test our implementation. The results are as follows.

I am looking forway to see you soon.
i am looking to Norway to see you soon. - 99.7757%
i am looking forward to seeing you soon. - 99.7722%

The cat sat at mat.
the cat sat at the mat. - 99.8284%

The were angr .
they were anger. - 99.8171%
they were angry. - 99.8404%
they were engr. - 99.7457%

ACKNOWLEDGMENT

This work has been completed under the guidance of Prof. Uday Nayak, our project guide. We are thankful for his constant advice, guidance and encouragement that have helped us in completing this report successfully. We are also thankful to all the other faculty & staff members of our department for their kind co-operation and help.

REFERENCES

[1] H. Ng, S. Wu, Y. Wu, C. Hadiwinoto and J. Tetreault, "The CoNLL-2013 Shared Task on Grammatical Error Correction," Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pages 1–12, Sofia, Bulgaria, August 8-9 2013

[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "B.E.R.T: Pre-training of Deep Bidirectional Transformers for Language Understanding": arXiv:1810.04805v2 [cs.CL] 24 May 2019.

[3] Vaswani, et al, "Attention Is All You Need": arXiv:1706.03762v5 [cs.CL] 6 Dec 2017

[4] R. Horev, "B.E.R.T Explained: State of the art language model for NLP": https://towardsdatascience.com/B.E.R.T-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

[5] Huggingface, "Pytorch Transformers": https://github.com/huggingface/pytorch-transformers

[6] Bryant and T. Briscoe, "Language Model Based Grammatical Error Correction without Annotated Training Data": Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 247–253

[7] Bustamante, F. R., and León, F. S., "Gramcheck: A grammar and style checker," in Proceedings of the 16th conference on Computational linguistics-Volume 1 (Association for Computational Linguistics). pp. 175–181, 1996.

[8] Ghosalkar, P., Malagi, S., Nagda, V., Mehta, Y., & Kulkarni, P,. English Grammar Checker, 2016.

[9] Han, N.-R., Chodorow, M., and Leacock, C., "Detecting errors in english article usage with a maximum entropy classifier trained on a large, diverse corpus.." in LREC, 2004.

[10] Rozovskaya, A., Chang, K.-W., Sammons, M., and Roth, D., "The university of illinois system in the conll-2013 shared task," in Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pp. 13–19, 2013.

[11] Yuan, Z., and Briscoe, T., "Grammatical error correction using neural machine translation" in HLT-NAACL, pp. 380–386, 2016.

[12] Chollampatt, S. and Ng, H.T., April "A multilayer convolutional encoder-decoder neural network for grammatical error correction." In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[13] Napoles, C., Sakaguchi, K., Post, M., and Tetreault, J., "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, Vol. 2, pp. 588–593, 2015.