# Project Proposal

Machine Learning for Language Understanding

Pranav Guntunur and Anand Tyagi

## I *Motivation*

Grammatical Error Correction (GEC) aims to correct different kinds of grammatical errors in text, including spelling, grammar and word choice. Multiple types of models have achieved high $F_{0.5}$ scores in GEC tasks, from sequence-to-sequence (seq2seq) models (76.88) [1] to unsupervised learning with transformers (69.34) [2]. Moreover, according to *Li et. al.* [3], "unsupervised models pre-trained on large corpora have boosted performance in many natural language processing tasks," upon which they utilized BERT [4] in order to perform GEC with minimal supervision. While BERT achieves a performance of more than 70%, we believe ELECTRA [5], which uses a different method of pre-training and performs better overall on the GLUE Benchmark [6], will be able to do better.

## II *Goals*

1. We want to see if ELECTRA's new pre-training method of Replaced Token Detection (RTD) proves to perform better on GEC. Specifically, we want to see how well ELECTRA performs at GEC compared to other existing models.
2. *Li et al* proposes several masking strategies — which involves choosing specific tokens to mask — for a Masked Language Model (MLM) to evaluate the performance of BERT-like models on GEC. Similarly, we want to try several replacement strategies on ELECTRA-like models, where we will use the data to guide which specific set of tokens to replace to see if this will improve ELECTRA's performance on GEC.

## III *Data*

Since we are focused on minimally supervised models, such as BERT and ELECTRA, we will be following BEA-2019 Shared Task's Low Resource Track (formerly Unsupervised Track) [7] which uses the Write&Improve+LOCNESS corpus (annotated by the Error Annotation Toolkit [8]). Since ELECTRA heavily notes that it is able to learn more effectively on less amount of data, having too little training data is not a problem we expect will hinder the performance of the model.

## IV *Tools*

The main tools we expect to be using for this project are the ELECTRA model, the code of which is provided on github by , and the Write&Improve+LOCNESS corpus, which is provided by the BEA shared task - 2019 dataset. We may also employ the use of NYU's HPC.

## V *Plan of Action*

1. We will familiarize ourselves with the current state-of-the-art GEC models and the ELECTRA model. We will also download the Write&Improve+LOCNESS corpus onto the server and make sure labels, annotations are what we need (verify data integrity and fit for our baseline task). All of this will be completed by April 6.
2. Implement a baseline system with ELECTRA and the corpus and see how well it performs on a simplified single-edit setting (attempting to correct sentences with a single error) in regards to the BEA Shared Task - 2019's Low-resource track's systems. This will be completed by April 13.
3. Implement and evaluate multiple token replacement strategies to enhance the pre-training task and see how ELECTRA performs with this addition in comparison to BERT. Since we expect the forming of the replacement strategies to take the majority of our time, we expect this task to take until May 1.
4. Finally, we will finish writing about our research, ablation tests, and findings from May 1 until the paper is due on May 13.

## VI *Collaboration Statement*

For this proposal, both Pranav and Anand each opened 3 individual papers and 2 shared papers to familiarize ourselves with the current GEC literature. For this write up specifically, Pranav wrote the

Motivation and Data sections and Anand wrote the Goals and Plan of Action section. However, the entire paper was looked over and revised by both individuals together.

For the project, both individuals will read all of the same papers and documentation in order to understand the tools and data. Both Pranav and Anand have similar ML and NLP experience, but Pranav is more knowledgeable about handling language data while Anand is more familiar with the linguistics side of NLP. The similar level of mathematics and coding skill, however, will allow both individuals to contribute equally to the implementation and testing of the code in the project.

## VII *References*

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

**[2]** Grundkiewicz, Roman, Marcin Junczys-Dowmunt, and Kenneth Heafield. "Neural grammatical error correction systems with unsupervised pre-training on synthetic data." *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications.* 2019.

**[3]** Li, Yiyuan, Antonios Anastasopoulos, and Alan W. Black. "Towards Minimal Supervision BERT-based Grammar Error Correction." *arXiv preprint arXiv:2001.03521* (2020).

**[4]** Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

**[5]** Clark, Kevin, et al. "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators." *International Conference on Learning Representations.* 2019.

**[6]** Wang, Alex, et al. "Glue: A multi-task benchmark and analysis platform for natural language understanding." *arXiv preprint arXiv:1804.07461* (2018).

**[7]** Bryant, Christopher, et al. "The BEA-2019 shared task on grammatical error correction." *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*. 2019.

**[8]** Bryant, Christopher, Mariano Felice, and Edward Briscoe. "Automatic annotation and evaluation of error types for grammatical error correction." Association for Computational Linguistics, 2017.