Multilingual Grammatical Error Detection

Wesley Scivetti[†], **Ulie Xu**[†], **Neuman Alkhalil** [‡], **Didier Akilimali** [‡]
Georgetown University
[†]Department of Linguistics, [‡]Department of Computer Science
{wss37, lx59, na892, dda34}@georgetown.edu

Abstract

Most recent work on Grammatical Error Detection (GED) has only considered data from English. This paper investigates the capabilities of pretrained language models for GED in 5 languages. Following recent work on GED, we utilize a fine-tuning approach for token classification for all of our experiments. We fine-tune monolingual BERT and ELECTRA models in each of the 5 languages, and also fine-tune mBERT and XLM-Roberta using the combined data from all 5 languages. We find that ELECTRA outperforms BERT consistently, and that the jointly fine-tuned XLM-Roberta outperforms monolingual models in 3 of the 5 languages we investigate. Error analysis in English indicates that errors in form are easiest to predict, while subtle errors in word usages are still difficult for all of the models we fine-tuned.

1 Introduction

Learning a language inevitably involves making mistakes. Getting accurate corrections for past errors can be an invaluable experience for language learners, but it is time consuming for language experts to identify errors and provide such corrections. Thus, automatic systems for Grammatical Error Detection (GED) and Grammatical Error Correction (GEC) have considerable pedagogical potential as aids for language learners.

While the usefulness of automatic GED systems is obvious, there are several considerations which make this task difficult. Firstly, there needs to be available data annotated with errors in order to train supervised systems. In typical GED datasets, each word in a sentence is annotated as either *correct* or *incorrect*, though some more fine-grained labels are sometimes present, and some datasets annotate spans or entire sentences instead of tokens. Regardless of the exact annotation schema, datasets are typically imbalanced, with far more *correct* tokens than *incorrect* tokens. This is simply because learners typically make one or a few mistakes per sentence, while sentences which are mostly errors are rarely produced. To make the task even more challenging, some errors involve missing tokens or incorrectly ordered token sequences. In such cases, the typical strategy of labeling each token is somewhat limited.

The task of GED has been a topic of computational research for several decades, and a variety of architectures have been utilized. While Bidirectional LSTMs were the standard for several years, recent advances in the capabilities of Large Language Models (e.g. BERT, Devlin et al. 2019) have led to increases in the state-of-the-art performance. In particular, fine-tuning an ELECTRA (Clark et al., 2020) model has been shown to be particularly effective for the GED task (Yuan et al., 2021). Regardless of the architecture, the task is typically formulated as a binary token classification task, though other setups are possible.

The vast majority of previous literature has focused on GED in English. In particular, recent work with LLMs has shown great performance increases in English, but it is unclear how effective these techniques will be in other languages which may not have as robust pretrained language models trained in them. Furthermore, the potential for cross-lingual or multilingual grammatical error detection systems remains underexplored, which has led to the recent Multi-GED 2023 shared task[1].

This work helps address the gap in multilingual grammar error detection research by investigating the performance of both monolingual and multilingual large language models across five languages. In the monolingual setting, we fine-tune and compare versions of BERT and ELECTRA that have been pretrained in each of the 5 languages. In the multilingual setting, we fine-tune mBERT and XLM-Roberta (Conneau et al., 2020) on the combined data from all 5 languages. In the following

[1]https://github.com/spraakbanken/multiged-2023

Section, we outline previous work on GED, focusing in particular on recent work using large language models. In Section 3, we describe the data sources that were used for this study. In Section 5 we present our results in each of the 5 languages for our tested models. In Section 6 we analyze some notable errors from our system outputs. In Section 7 we discuss the limitations of our work, and in Section 8 we offer final reflections and conclusions from this study.

2 Related Work

Pre-neural attempts at automatic GED primarily relied on rule-based systems (Foster and Vogel, 2004) or maximum entropy classifiers using hand-crafted features (Chodorow et al., 2007; De Felice and Pulman, 2008). With the increased capabilities of neural networks and pretrained word embeddings, Rei and Yannakoudakis (2016) achieved state-of-the-art performance on the task using neural classification architectures with pretrained word2vec embeddings (Mikolov et al., 2013) as the input. They experiment with CNNs, RNNs, Bidirectional RNNs, LSTMs, and Bidirectional LSTMs, finding that a 2 layer Bi-LSTM leads to best performance.

Rei et al. (2017) expand upon the previous study by including artificially generated errors to augment the available training data. They formulate the task of error augmentation as a machine-translation (MT) task, with correct and incorrect sentences framed as the "source" and "target" languages respectively, and utilize a statistical machine translation (SMT) for the task. They also experiment with pattern-based methods of generating errors similar to the most common errors of learners. They find that using these two data augmentation approaches leads to performance improvements on GED when using a similar Bi-LSTM architecture similar to that proposed by Rei and Yannakoudakis (2016). In a related vein of work Kasewa et al. (2018) also attempt to augment GED training data using a MT approach. Unlike Rei et al. (2017), Kasewa et al. (2018) utilize a neural sequence-to-sequence system with attention for the error generation task. The improvements in the MT architecture lead to further performance gains, though they still utilize a Bi-LSTM architecture with word2vec embeddings.

Bell et al. (2019) present the first GED system which utilizes contextual embeddings.They experiment with concatenating ElMo, BERT, and Flair embeddings to pretrained word2vec embeddings, and using the concatenations as inputs to a Bi-LSTM architecture. Of the embedding types investigated, they find that BERT embeddings generally perform best, and concatenate the hidden representations from the last 4 layers of BERT onto the static embeddings. Additionally, they add a language modeling task, which they optimize their system for jointly alongside the GED task. They find that adding the language modeling objective allows the model to learn some "general" features which aid in the GED task. They also utilize a weighting hyperparameter for the loss functions of the two tasks, so that the combined loss function puts considerably more weight on the GED task. Using contextual embeddings alonside the LM auxiliary task leads to state-of-the-art performance, outpacing previous methods which solely utilize static embeddings.

While most of the above literature has focused solely on the task of GED, Yuan et al. (2021) combine state-of-the-art GED methodologies with a transformer based GEC system. Regarding GED, they are able to achieve state of the art performance by simply fine-tuning ELECTRA Clark et al. (2020) for the task, which outperforms BERT and XLNet (Yang et al., 2019) for this task. Unlike most previous work, which has considered GED as primarily a binary token classification task, the authors also test the capabilities for ELECTRA for detecting more fine-grained distinctions between types of errors. They test ELECTRA's capabilities on a GED task in 4-class, 25-class, and 55-class settings, where the number of classes are the number of different types of errors. While predictably F-.5 score drops as the number of classes increases, they still find that ELECTRA performs remarkably well in the multiclass setting. After experimenting with GED in the binary and multiclass settings, they then test if GED systems can help improve transformer based GEC systems. They utilize their GED system as an encoder in addition to a standard transformer architecture, finding that a system which is fine-tuned with the addition of the GED encoder outperforms a transformer baseline. For the present study, we take the lead of Yuan et al. (2021) in utilizing ELECTRA as a monolingual model for GED, since it was shown to outperform both BERT and XLNet.

Kaneko and Komachi (2019) extend upon the idea of using contextual embeddings for GED.

Lang	Tokens	Correct	Incorrect	Sentences
Czech	66351	52969	13382	29795
English	91008	82279	8729	28357
German	60772	51610	9162	19239
Italian	16464	13852	2612	6394
Swedish	23244	18898	4346	6729

Table 1: Information on the number of tokens, distribution of correct and incorrect tokens, and number of sentences for each language's dataset.

However, instead of using embeddings as input to a Bi-LSTM, they opt to fine-tune various BERT architectures directly. Unlike Bell et al. (2019), who utilize the hidden representations from the last 4 layers of BERT, Kaneko and Komachi (2019) propose and fine-tune a Multi-Head, Multi-Layer Attention (MHMLA) system, which learns attention weights for the representations of each layer as part of the fine-tuning process. They compare their MHMLA system to a standard BERT architecture, and a setup where the average of all hidden layer representations is fed into the output classification layer. They find that the MHMLA outperforms the other versions of BERT, though the difference is not drastic.

In contrast to the previous works, which have focused on GED for English, Cheng and Duan (2020) investigate using pretrained language models for GED in Chinese. They utilize monolingual Chinese versions of BERT and RoBERTa, finding that BERT performs slightly better. In contrast to previous work, which frames GED as a token classification task, Cheng and Duan (2020) instead consider it as a sequence classification task, and produce one label ("correct" or "incorrect") per sentence. Cheng and Duan (2020) is important context for the present study because it is one of few endeavors into non-English GED, but still does not investigate GED in a multilingual setting.

3 Data

The datasets used in this project were provided as part of a shared task on GED acoss five different languages. Datasets came from monolingual learner corpora in the following languages: Czech (Náplava et al., 2022), English (Yannakoudakis et al., 2011; Kuzmenko and Kutuzov, 2014), German (Boyd et al., 2014), Italian (Boyd et al., 2014), and Swedish (Volodina et al., 2019). The datasets provided consist of a total of 1,735,266 tokens, with 251,573 errors across 117,413 sentences, giving us an average error rate of 0.1450 (4 d.p.).

The datasets are split into the 5 different languages, with each language having files containing a training set, a dev set, and a test set. Both the training and dev sets are labeled with "c" and "i" to identify errors in the sentences, however the test sets are unlabeled.

Due to the test sets being unlabeled, we chose to not use it as it does not allow for evaluation of model testing since there is no gold label to compare to the predicted label, and as such a decision was made to take the dev set as the test set, and split the training set with 80 percent being used for training, and 20 percent being used as a dev set. This left us with the following data composition in Table 1

4 Methodology

We experimented with four different models: BERT and mBERT (Devlin et al., 2019), XLM-RoBERTa (Conneau et al., 2020), and ELECTRA (Clark et al., 2020). Since the four models are encoder models, we had to add and finetune a token classification layer on top of each pre-trained encoder model. Thanks to Hugging Face we did not have to write the code adding the classification layer on top of our individual models. We imported the four pre-trained models from Hugging Face's transformer library with the classification layer already on top. Being monolingual models, BERT and ELECTRA were finetuned on each language separately. On the other hand, the two multilingual models were finetuned on the combined data of all languages. Training and development data for the five languages were obtained from the Multilingual Grammatical Error Detection Task (MultiGED-2023) GitHub repository.

5 Results

Evaluation was carried out in terms of token-based accuracy, precision, recall, and F0.5 scores to be

Language	Model	Accuracy	Precision	Recall	F0.5
Czech	BERT	.9050	.8688	.6231	.8053
	ELECTRA	.8972	**.8944**	.5559	.7973
	mBERT	.8983	.8543	.5974	.7867
	XLM-RoBERTa	**.9086**	.8551	**.6585**	**.8069**
English	BERT	.92	.73	.33	.59
	ELECTRA	**.9326**	**.7564**	**.4379**	**.6603**
	mBERT	.9259	.7427	.3475	.6050
	XLM-RoBERTa	.9291	.7160	.4329	.6332
German	BERT	.9054	.8072	.4893	.7144
	ELECTRA	.9198	**.8288**	.5898	.7667
	mBERT	.9155	.7982	.5881	.7450
	XLM-RoBERTa	**.9297**	.8277	**.6737**	**.7915**
Italian	BERT	.88	.71	.46	.64
	ELECTRA	**.9206**	**.8464**	.6099	**.7855**
	mBERT	.9037	.8025	.5211	.7242
	XLM-RoBERTa	.9181	.8250	**.6137**	.7719
Swedish	BERT	.8713	.7678	.4466	.6713
	ELECTRA	.8510	**.7952**	.2734	.5755
	mBERT	.8659	.7660	.4075	.6514
	XLM-RoBERTa	**.8898**	.7899	**.5596**	**.7298**

Table 2: The accuracy, precision, recall, and F0.5 scores of 4 models on all 5 languages for the development dataset. Best performing model statistics are in bold.

consistent with previous grammatical error detection work. F0.5 score was adopted instead of F1 because humans judge false positives more harshly than false negatives, so precision was weighed more importantly than recall. Overall, evaluating on the development split that we sectioned from the training data, we found that ELECTRA and XLM-RoBERTa outperformed the other two models respectively in the monolingual and multilingual settings. In other words, ELECTRA beat monolingual BERT in the monolingual model category while XLM-RoBERTa beat multilingual BERT in the multilingual model category. ELECTRA outperformed every model in precision in every language, which means that it is the most reliable model in correctly classifying positive instances and minimizing the occurrences of false positives. However, it was still to our surprise that XLM-RoBERTa even outperformed ELECTRA in languages such as Czech and Swedish because multilingual models' training procedure of mixing all language data together made us hypothesize that monolingual models would have outperformed multilingual models. Table 2 shows the accuracy, precision, recall, and F0.5 scores of the 4 models in each language, and figure 1 shows the F0.5 scores in a grouped bar plot.

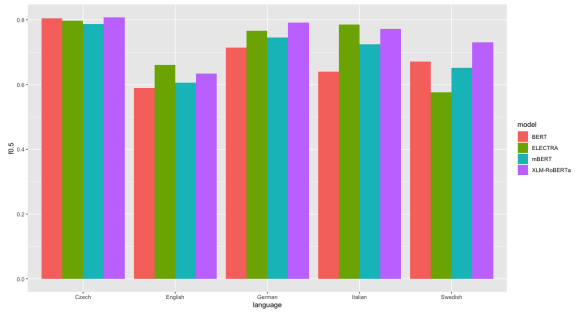

Figure 1: A grouped bar plot of the F0.5 scores of all 4 models on all 5 languages on the development dataset.

After evaluating on the models with the accuracy, precision, recall, and F0.5 scores, we proceeded to choose ELECTRA as our best-performing model for English and Italian, and XLM-RoBERTa as our best-performing model for Czech, German, and Swedish, and tested the two models on the languages that they respectively performed the best in. Table 3 shows a comparison between our test statistics with the competition results of the three top current (as of May 12, 2023) submission groups on the MultiGED-2023 GitHub page on Czech, English, and German, and table 4 shows Italian and Swedish. Our test results are pretty competitive

	Czech			English			German		
Team name	**P**	**R**	**F0.5**	**P**	**R**	**F0.5**	**P**	**R**	**F0.5**
Our team	.8384	.4872	.7328	.7657	.4098	.6524	.8237	.6583	.7843
EliCoDe	.8201	.5179	.7344	.7364	.5034	.6740	.8478	.7375	.8232
DSL-MIM-HUS	.5831	.5569	.5776	.7236	.3781	.6118	.7780	.5192	.7075
Brainstorm Thinkers	.6235	.2344	.4681	.7021	.3755	.5981	.7794	.4755	.6911

Table 3: The precision, recall, and F0.5 scores on Czech, English and German between our performance and the 3 top competition teams.

	Italian			Swedish		
Team name	**P**	**R**	**F0.5**	**P**	**R**	**F0.5**
Our team	.8418	.6457	.7936	.8105	.5242	.7307
EliCoDe	.8667	.6796	.8215	.8180	.6634	.7816
DSL-MIM-HUS	.7572	.3867	.6355	.7485	.4492	.6605
Brainstorm Thinkers	.7065	.3646	.5949	.7381	.3994	.6311

Table 4: The precision, recall, and F0.5 scores on Italian and Swedish between our performance and the 3 top competition teams.

compared to the top teams' performances; however, it is also important to re-emphasize that we used the development data provided from the GitHub repository as our test data, so the comparison needs to be interpreted with caution.

6 Discussion

In general, we find that error detection is generally easiest for spelling and token level errors and hardest for semantic errors. Syntactic errors, especially agreement errors, are often detected, but are sometimes missed, particularly by BERT. Below, we show a few examples of errors in our model predictions. For the following examples, words colored in **red** were errors that the models *failed* to identify (in other words, the models marked these tokens correct). The words in **green** are words that at least one of the models *succeeded* in identifying as incorrect.

(1) The weekend **started** and **this** Friday we went to the **dyscoteque**.

In Example (1), we can see that all the models fail to detect the first two errors, which we consider to be more semantic. It is strange to say *this Friday* when talking about a past event, though we are less sure about what the error is with *started*. This is one case where the data itself may be incorrect. However, all the models were able to annotate *dyscoteque* as incorrect, as it is misspelled.

(2) But I think that the most important **is** my cellular phone: I can **give** a call or receive a call from **everywhere**.

In Example (2), we can see that none of the errors are correctly identified by any model. We again view these errors as semantic/pragmatic primarily. Syntactically, the sentence is generally well formed, and the semantic errors are still plausible. For example *giving* a phone call is not completely nonsensical, it is just not a preferred way of expressing the intended meaning in English. Similarly, making a call from *everywhere* instead of *anywhere* is understandable, but a slightly incorrect usage of the word. These errors seem quite subtle, and demonstrate the challenging aspects of this task.

(3) The problem **about** this job is that you have to deal with huge **amounts** of people.

In Example (3), our ELECTRA English model correctly identified the errors, while BERT and the multilingual models (mBERT, XLM-Roberta) did not. These errors seem to fall in between the boundary of syntax and semantics. Using *amounts* with a mass noun like *people* is technically a syntactic error, albeit a minor one. Similarly, not recognizing that *problem* attracts the preposition *with* and not *about* mostly comes down to the syntactic idiosyncrasies of *problem*. We are not surprised that the models struggle with this example, but it is interesting that ELECTRA alone is successful. Perhaps ELECTRA's pretraining GAN task is more advan-

tageous than Masked Language Modeling in the context of GED.

7 Limitations

Despite the promising results of the project, there are some limitations that need to be taken into consideration. Firstly, the data used in this project is limited to only five languages. Although the data covers some of the most commonly used languages, it may not be sufficient for evaluating the performance of the models on other languages.

Additionally, the datasets is relatively small in size. This may limit the model's ability to generalize to new and unseen data, especially to other languages not present in the datasets.

Moreover, the test datasets provided are unlabeled which do not allow us to obtain any metrics when testing them as we do not have the gold labels to compare to the predicted labels. Because of this, the training set used was further reduced by an additional 20 percent as 80 percent was used for training, the remainder of the set was used as a dev set, and the dev set provided was used as the test set.Furthermore, the datasets all have a relatively low error rate. This can affect the models ability to adapt to the task as it does not have many examples of incorrect labels to learn from.

Lastly, the models used in this project are based on pre-trained language models. These pre-trained language models are trained on large datasets of text, and the language patterns and structures learned by the model may be influenced by the language distribution in the training data. This may lead to biases in the model's performance, particularly when applied to languages not represented in the training data.

8 Reflection

This project provided a valuable opportunity to learn about the design decisions, trial and error, and experimental comparisons involved in developing models for GED using pre-trained language models.

One of the key design decisions made in this project was the model selection process. We had the option of doing 5 monolingual models or 1 multilingual model, and given we had more resources among the 4 of us we chose to use 2 sets of monolingual models and 2 multilingual models provided by the Hugging Face transformers library and do a comparison of them. The models we chose were BERT, ELECTRA, mBERT, and XLM-RoBERTa, each of which have shown promising results in a range of NLP tasks. Each of the models were fine-tuned on the five different languages, and through experimental comparison, we found that the multilingual models generally outperformed the monolingual models on the languages they were trained on. Additionally, we tried several models that we had to abandon due to excessive memory usages and training time, including mBART and XLM-RoBERTa large.

Another design decision made was to split the training set into a training and dev set (80/20 split), and use the dev set provided as a test set. The reason behind this is to combat the problem where the test sets provided were unlabeled, therefore not giving us the means to evaluate the model's performance as we cannot compute whether the predicted labels were correct or not as there is no gold label to compare to.

One design decision which is to be reflected upon is the decision to shuffle the sentences in the datasets. This was done in order to randomize the datasets and give the models a more difficult time to find patterns in the sentences as it removes the context, furthering the need for the models to build more complex relationships and become more robust. This in hindsight, however, might have been counter intuitive as the size of the datasets were small, therefore not allowing the model to have enough information to find these patterns. As shown in (1) where there are 2 semantic errors which are hard to determine without any context, even for humans. So in situations like this, context would be beneficial. In the future, we could explore feeding larger chunks into models, instead of sentences in isolation. This would require more thought in how to properly segment the data, but might lead to increased performance.

Another direction we could go if we had more time would be to try these methods in combination with some of the data augmentation methods from previous work. Past work (Rei et al., 2017; Kasewa et al., 2018) showed that using MT setups to create artificial error examples led to increased performance on GED, but this technique was not used in the most recent work of Yuan et al. (2021). It could be that these augmentation techniques would benefit fine-tuning ELECTRA and XLM-RoBERTa like they did previous systems. Additionally, data augmentation may be more effective as the state-

of-the-art for sequence to sequence modeling has advanced.

In conclusion, this project provided valuable insights into the design decisions, trial and error, and experimental comparisons involved in developing and fine-tuning multiple pre-trained models for GED across 5 languages. Through our work, we learned about the importance of the different decisions made in the project. These insights are critical for advancing the field of NLP and developing more accurate and effective models for a range of tasks and languages.

References

Samuel Bell, Helen Yannakoudakis, and Marek Rei. 2019. Context is Key: Grammatical Error Detection with Contextual Word Representations. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, page 103–115, Florence, Italy. Association for Computational Linguistics.

Adriane Boyd, Jirka Hana, Lionel Nicolas, Detmar Meurers, Katrin Wisniewski, Andrea Abel, Karin Schöne, Barbora Štindlová, and Chiara Vettori. 2014. The MERLIN corpus: Learner language and the CEFR. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1281–1288, Reykjavik, Iceland. European Language Resources Association (ELRA).

Yong Cheng and Mofan Duan. 2020. Chinese Grammatical Error Detection Based on BERT Model. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, page 108–113, Suzhou, China. Association for Computational Linguistics.

Martin Chodorow, Joel Tetreault, and Na-Rae Han. 2007. Detection of grammatical errors involving prepositions. In *Proceedings of the Fourth ACL-SIGSEM Workshop on Prepositions*, pages 25–30, Prague, Czech Republic. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, and Quoc V Le. 2020. Electra: Pre-training text encoders as discriminators rather than generators.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 8440–8451, Online. Association for Computational Linguistics.

Rachele De Felice and Stephen G. Pulman. 2008. A classifier-based approach to preposition and determiner error correction in L2 English. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 169–176, Manchester, UK. Coling 2008 Organizing Committee.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jennifer Foster and Carl Vogel. 2004. Parsing ill-formed text using an error grammar. *Artificial Intelligence Review*, 21:269–291.

Masahiro Kaneko and Mamoru Komachi. 2019. Multi-Head Multi-Layer Attention to Deep Language Representations for Grammatical Error Detection. *Computación y Sistemas*, 23(3).

Sudhanshu Kasewa, Pontus Stenetorp, and Sebastian Riedel. 2018. Wronging a Right: Generating Better Errors to Improve Grammatical Error Detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, page 4977–4983, Brussels, Belgium. Association for Computational Linguistics.

Elizaveta Kuzmenko and Andrey Kutuzov. 2014. Russian error-annotated learner English corpus: a tool for computer-assisted language learning. In *Proceedings of the third workshop on NLP for computer-assisted language learning*, pages 87–97, Uppsala, Sweden. LiU Electronic Press.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Jakub Náplava, Milan Straka, Jana Straková, and Alexandr Rosen. 2022. Czech Grammar Error Correction with a Large and Diverse Corpus. *Transactions of the Association for Computational Linguistics*, 10:452–467.

Marek Rei, Mariano Felice, Zheng Yuan, and Ted Briscoe. 2017. Artificial error generation with machine translation and syntactic patterns. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 287–292, Copenhagen, Denmark. Association for Computational Linguistics.

Marek Rei and Helen Yannakoudakis. 2016. Compositional sequence labeling models for error detection in learner writing. In *Proceedings of the 54th Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1181–1191, Berlin, Germany. Association for Computational Linguistics.

Elena Volodina, Lena Granstedt, Arild Matsson, Beáta Megyesi, Ildikó Pilán, Julia Prentice, Dan Rosén, Lisa Rudebeck, Carl-Johan Schenström, Gunlög Sundberg, and Mats Wirén. 2019. The SweLL Language Learner Corpus: From Design to Annotation. *Northern European Journal of Language Technology*, 6:67–104.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

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*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

Zheng Yuan, Shiva Taslimipoor, Christopher Davis, and Christopher Bryant. 2021. Multi-Class Grammatical Error Detection for Correction: A Tale of Two Systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, page 8722–8736, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Appendix

Our GitHub repository is attached here for review.