Two Neural Models for Multilingual Grammatical Error Detection

The Quyen Ngo and **Thi Minh Huyen Nguyen** and **Phuong Le-Hong***✉
VNU University of Science, Vietnam National University, Hanoi
FPT Technology Research Institute, FPT University, Hanoi*
(ngoquyenbg|huyenntm|phuonglh)@vnu.edu.vn

Abstract

This paper presents two neural models for multilingual grammatical error detection and their results in the MultiGED-2023 shared task. The first model uses a simple, purely supervised character-based approach. The second model uses a large language model which is pretrained on 100 different languages and fine-tuned on the provided datasets of the shared task. Despite simple approaches, the two systems achieved promising results. One system has the second best F-score; the other is in the top four of participating systems.

1 Introduction

Grammatical Error Detection (GED) is the task of detecting different kinds of errors in text such as spelling, punctuation, grammatical, and word choice errors. It is one of the key components in the grammatical error correction (GEC) community. This paper concerns with the development of different methods for subtoken representation and their evaluation on standard benchmarks for multiple languages. Our work is inspired by the recent shared task MultiGED-2023. The aim of this task is to detect tokens in need of correction across five different languages, labeling them as either correct ("*c*") or incorrect ("*i*"), i.e. performing binary classification at the token level.

Recent GED methods make use of neural sequence labeling models, either recurrent neural networks or transformers. The first experiments using convolutional neural network and long short-term memory networks (LSTM) models for GED was proposed in 2016 (Rei and Yannakoudakis, 2016). Later, a bidirectional, attentional LSTM was used to jointly learn token-level and sentence-level representations and combine them so as to detect grammatically incorrect sentences and to identify the location of the error tokens at the same time (Rei and Søgaard, 2019). The bidirectional LSTM model was also used together with grammaticality-specific word embeddings to improve GED performance (Kaneko et al., 2017). A bidirectional LSTM model was trained on synthetic data generated by an attentional sequence-to-sequence model to push GED score (Kasewa et al., 2018). Best-performing GED systems employ transformer block-based model for token-level labeling. A pretrained BERT model has been fine-tuned for GED and shown its superior performance in (Kaneko and Komachi, 2019). The BERT model has also been shown significant improvement over LSTM models in both GED and GEC (Liu et al., 2021). The state-of-the-art GED method uses a multi-class detection method (Yuan et al., 2021).

In this work, we also employ state-of-the-art sequence labeling methods, which are based on LSTM or BERT. In contrast to previous work, we focus on different representations of tokens at subtoken levels. Our best-performing system can process multiple languages using a single model.

2 Methods

We use two different token representations, one at the character level, and one at the subtoken level.

2.1 Character-based Representation

In this representation, the j-th input token of a sentence is represented by the concatenation of three vectors (b_j, m_j, e_j) corresponding to its characters. More precisely, the token is represented by vector $x_j = (b_j, m_j, e_j)$ where the first vector b_j and the third vectors e_j represent the first and last character of the token respectively. The second vector m_j represents *a bag of characters* of the middle subtoken without the initial and final positions.

Phuong Le-Hong, The Quyen Ngo and Thi Minh Huyen Nguyen. Two Neural Models for Multilingual Grammatical Error Detection. *Proceedings of the 12th Workshop on Natural Language Processing for Computer Assisted Language Learning (NLP4CALL 2023)*. Linköping Electronic Conference Proceedings 197: 40–44.

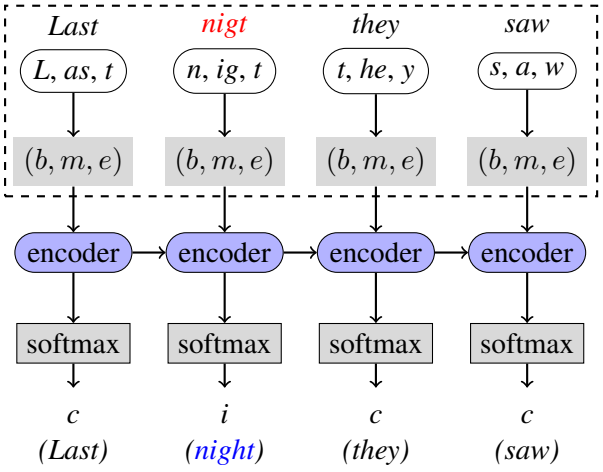

Figure 1: Our character-based model

The dotted frame in Figure 1 depicts this representation. For example, the token "*Last*" is represented as a concatenation of the following vectors: (1) an one-hot vector for character *L*; (2) an one-hot vector for character *t*, and (3) a bag-of-character multihot vector for the internal characters *a, s*. Thus, each token is represented by a vector of size $3V$ where V is the size of the alphabet. The label y_j is predicted by a softmax layer:

$$y_j = \frac{\exp(W_j \cdot h_j)}{\sum_k \exp(W_k \cdot h_j)}.$$

This representation is inspired by a semi-character word recognition method which was proposed by Sakaguchi et al. (2017). It was demonstrated that this method is significantly more robust in word spelling correction compared to character-based convolutional networks.

2.2 Subtoken-based Representation

Recent language processing systems have used unsupervised text tokenizer and detokenizer so as to make a purely end-to-end system that does not depend on language-specific pre- and post-processing. SentencePiece is a method which implements subword units, e.g., byte-pair-encoding – BPE (Sennrich et al., 2016) and unigram language model (Kudo, 2018) with the extension of direct training from raw sentences. Using this method, the vocabulary size is predetermined prior to the neural encoder training. Our system also uses subtoken representation.

2.3 LSTM and BERT Encoders

The LSTM network is a common type of recurrent neural networks which is capable of processing sequential data efficiently. This was a common method prior to 2017, before Transformers (Vaswani et al., 2017), which dispense entirely with recurrence and rely solely on the attention mechanism. Despite being outdated, we developed a purely supervised LSTM encoder to test the effectiveness of the character-based method.

We employ the XLM-RoBERTa model as another encoder in our system. RoBERTa (Liu et al., 2019) is based on Google's BERT model released in 2018 (Devlin et al., 2019). It modifies key hyperparameters, removing the next-sentence pretraining objective and training with much larger mini-batches and learning rates. RoBERTa has the same architecture as BERT, but uses a byte-level BPE as a tokenizer. The XLM-RoBERTa model was proposed in 2020 (Conneau et al., 2020), which is based on RoBERTa. It is a large multilingual language model, trained on 100 languages, 2.5TB of filtered CommonCrawl data. It has been shown that pretraining multilingual models at scale leads to significant performance gains for a wide range of cross-lingual transfer tasks. Unlike some XLM multilingual models, this model does not require language tensors to understand which language is used, and should be able to determine the correct language from the input ids.

3 Experiments

This section presents the datasets in use, experimental settings and obtained results of our system.

3.1 Datasets

The datasets are provided by the MultiGED-2023 shared task.[1] The shared task provides training, development and test data for each of the five languages: Czech, English, German, Italian and Swedish. The training and development datasets are available in the MultiGED-2023 GitHub repository, and test sets are released during the test phase for participating teams. Table 1 shows the statistics of the datasets.

3.2 Evaluation Metric

Evaluation is carried out in terms of token-based precision, recall and $F_{0.5}$, consistent with previous work on error detection. $F_{0.5}$ is used instead of F_1 because humans judge false positives more harshly than false negatives and so precision is more important than recall.

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

Lang.	Sents.	Tokens	Errors	Rate
Czech	35,453	399,742	84,041	0.210
English	33,243	531,416	50,860	0.096
German	24,079	381,134	57,897	0.152
Italian	7,949	99,698	14,893	0.149
Swedish	8,553	145,507	27,274	0.187

Table 1: Statistics of datasets in five languages

Language	Precision	Recall	$F_{0.5}$
Czech	34.93	**63.95**	38.42
English (FCE)	20.76	29.53	22.07
English (REA)	–	–	–
German	25.18	44.27	27.56
Italian	25.79	44.24	28.14
Swedish	26.40	55.00	29.46

Table 2: Performance of the VLP-char system on the private test set. The number in bold font is the best recall of all participating systems on the Czech dataset.

3.3 Experimental Settings

Our first system, namely VLP-char, uses the character-based token representation and the LSTM encoder. Its parameters are initialized with random vectors in each run. This allows us to establish results in a pure supervised learning setting rather than a semi-supervised or transfer learning setting. The same model is trained separately for each language, resulting five models. All five language-specific models are trained with the Adam optimizer (Kingma and Ba, 2015), and with learning rate 5×10^{-4}. We use the cross-entropy loss function for multinomial classification as usual. All models are trained in 80 epochs. The maximum sequence length is set to 60 tokens – this is enough to cover most sentences in the provided datasets. Since the data is highly imbalanced – the error rates are from only 10% (for English) to 24% (for Czech), we set the incorrect label weight to 90% and the correct label weight to 10% when computing the objective function.

This system does not use any external resources; only datasets provided by the organizers are used to train and validate the models. We use the BigDL library[2] as the deep learning framework. Our code is publicly available on GitHub.[3]

Our second system, namely DSL-MIM-HUS, uses the subtoken-based representation and the pretrained XLM-RoBERTa embeddings.[4] This system uses the library NERDA[5] to fine-tune the pretrained embeddings on all datasets. That is, we combine all the provided datasets (training and development splits) into one large dataset and perform the experiment on this combined one. There is thus only one model for all the five languages. The combined dataset is divided into training, development and test split with the ratios 0.8, 0.1 and 0.1, respectively. There are 82,976 training samples, 10,371 development samples and 10,371 test samples respectively. We did not keep the proportion of different language data the same when sampling. It had been more beneficial if the proportion would have been kept since the sizes of languages are very different – there are three times more German sentences than Italian sentences. The hyperparameters are tuned on the development set and selected as follows: the learning rate of 10^{-5}, the number of training epochs of 20.

3.4 Results

3.4.1 Supervised System

Without using any external datasets or pretrained embeddings, the VLP-char system obtained mediocre results. It ranks the fourth place among participating systems. This sytem consistently gives higher recall than precision on all the languages, while other systems have better precision than recall. It achieves 63.95% of recall on the Czech test set, which is the highest recall among participating systems for this language, as shown in Table 2.

Despite mediocre results, this system represents what we can build with very limited data.

3.4.2 Pretrained System

On our test split, the system DSL-MIM-HUS achieves a precision of 80.88%, a recall of 64.07% and $F_{0.5}$ of 71.50% for incorrect token prediction. The corresponding scores on the training set is 98.54%, 96.75%, and 97.64%, respectively. Since this combined dataset contains all the provided samples of all languages, it does not make sense to evaluate on each language separately.

On the private test set of the shared task MultiGED-2023 (Volodina et al., 2023), the system DSL-MIM-HUS is the second highest ranking. It achieves the best score among participating

[2] *https://github.com/intel-analytics/BigDL*
[3] *https://github.com/phuonglh/vlp/con/*
[4] *https://huggingface.co/xlm-roberta-large*
[5] *https://github.com/ebanalyse/NERDA*

Language	Precision	Recall	$F_{0.5}$
Czech	58.31	55.69	57.76
English (FCE)	72.36	37.81	61.18
English (REA)	62.81	28.88	**50.86**
German	77.80	51.92	70.75
Italian	75.72	38.67	63.55
Swedish	74.85	44.92	66.05

Table 3: Performance of the DSL-MIM-HUS system on the private test set. The number in bold font is the best score of all participating systems on the English REALEC dataset.

systems on the English REALEC dataset. Table 3 shows the performance of this system on the private test set, as announced by the organizers.

Although the XLM-RoBERTa system clearly outperformed the LSTM system, the LSTM system was trained on a fraction of the data available to the XLM-RoBERTa system.

4 Conclusion

We have presented two neural models for multilingual grammatical error detection and their results in the MultiGED-2023 shared task. One model uses a purely supervised LSTM network on a character-based token representation. The other model uses a pretrained BERT network on a subtoken representation. The two systems have achieved promising results in the shared task.

We are going to seek a better way to exploit syntactic and semantic information which comes from a dependency parser. We believe that explicit syntactic and semantic dependency between tokens of a sentence will be fruitful in detecting grammatical errors. In a recent study, we have demonstrated the usefulness of syntactic structures in improving lexical embeddings (Dang and Le-Hong, 2021). The idea of incorporating constituent-based syntax has also been shown effective for GED as well (Zhang and Li, 2022).

References

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 ACL*, pages 8440–8451, Online. ACL.

Hoang-Vu Dang and Phuong Le-Hong. 2021. A combined syntactic-semantic embedding model based on lexicalized tree-adjoining grammar. *Computer Speech and Language*, 68(2021):101202.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre–training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*, pages 1–16, Minnesota, USA.

Masahiro Kaneko and Mamoru Komachi. 2019. Multi-head multi-layer attention to deep language representations for grammatical error detection. *Computación y Sistemas*, 23(3):883–891.

Masahiro Kaneko, Yuya Sakaizawa, and Mamoru Komachi. 2017. Grammatical error detection using error- and grammaticality-specific word embeddings. In *Proceedings of the Eighth IJCNLP*, pages 40–48, Taipei, Taiwan. AFNLP.

Sudhanshu Kasewa, Pontus Stenetorp, and Sebastian Riedel. 2018. Wronging a right: Generating better errors to improve grammatical error detection. In *Proceedings of the 2018 EMNLP*, pages 4977–4983, Brussels, Belgium. ACL.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015*, pages 1–15, San Diego, CA, USA.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th ACL*, pages 66–75, Melbourne, Australia. ACL.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach.

Zhenghao Liu, Xiaoyuan Yi, Maosong Sun, Liner Yang, and Tat-Seng Chua. 2021. Neural quality estimation with multiple hypotheses for grammatical error correction. In *Proceedings of the 2021 NAACL*, pages 5441–5452, Online. ACL.

Marek Rei and Anders Søgaard. 2019. Jointly learning to label sentences and tokens. In *Proceeding of AAAI*, pages 6916–6923, Honolulu, Hawaii, USA.

Marek Rei and Helen Yannakoudakis. 2016. Compositional sequence labeling models for error detection in learner writing. In *Proceedings of the 54th ACL*, pages 1181–1191, Berlin, Germany. ACL.

Keisuke Sakaguchi, Kevin Duh, Matt Post, and Benjamin Van Durme. 2017. Robsut wrod reocginiton via semi-character recurrent neural network. In *Proceedings of the 31st AAAI*, AAAI'17, pages 3281–3287. AAAI Press.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th ACL*, pages 1715–1725, Berlin, Germany.

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

Elena Volodina, Christopher Bryant, Andrew Caines, Orphée De Clercq, Jennifer-Carmen Frey, Elizaveta Ershova, Alexandr Rosen, and Olga Vinogradova. 2023. MultiGED-2023 shared task at NLP4CALL: Multilingual Grammatical Error Detection. In *Proceedings of the 12th workshop on Natural Language Processing for Computer Assisted Language Learning (NLP4CALL)*, Tórshavn, Faroe Islands.

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 EMNLP*, pages 8722–8736, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yue Zhang and Zhenghua Li. 2022. CSynGEC: Incorporating constituent-based syntax for grammatical error correction with a tailored GEC-oriented parser.