The UNLP 2023 Shared Task on Grammatical Error Correction for Ukrainian

Oleksiy Syvokon
Microsoft
osyvokon@microsoft.com

Mariana Romanyshyn
Grammarly
mariana.romanyshyn@grammarly.com

Abstract

This paper presents the results of the UNLP 2023 shared task, the first Shared Task on Grammatical Error Correction for the Ukrainian language. The task included two tracks: GEC-only and GEC+Fluency. The dataset and evaluation scripts were provided to the participants, and the final results were evaluated on a hidden test set. Six teams submitted their solutions before the deadline, and four teams submitted papers that were accepted to appear in the UNLP workshop proceedings and are referred to in this report. The CodaLab leaderboard is left open for further submissions.

1 Introduction

Grammatical Error Correction (GEC) is an important task in natural language processing (NLP) that aims to automatically detect and correct grammatical errors in a given text. With the rapid growth of digital communication, GEC has become increasingly important in improving the quality of written communication. However, GEC is a complex task, especially for languages with complex grammar rules and rich morphology such as Ukrainian. Lack of large annotated and unlabeled datasets poses another challenge.

Shared tasks were a major contributing factor to the GEC progress in other languages: HOO-2011, HOO-2012, CoNLL-2013, CoNLL-2014, BEA-2019 (Dale and Kilgarriff, 2011; Dale et al., 2012; Ng et al., 2013, 2014; Mizumoto et al., 2012; Napoles et al., 2017; Bryant et al., 2019). Following that trend and to promote the development of GEC systems for Ukrainian, we organized the UNLP 2023 Shared Task on Grammatical Error Correction for Ukrainian. The shared task was organized as part of the Second Ukrainian NLP Workshop (UNLP 2023) colocated with EACL'2023.

The remainder of the paper is organized as follows. Section 2 describes the task. Section 3 describes the dataset. Section 4 explains how the submissions were evaluated. Finally, Section 5 presents the results of the participating teams.

2 Task description

The UNLP 2023 shared task required the participating systems to correct a text in the Ukrainian language to make it grammatical or both grammatical and fluent. Consequently, two tracks were suggested: GEC-only and GEC+Fluency. We made this distinction because fluency errors are more subjective and thus harder to correct.

In the GEC-only track, the participating systems were expected to correct grammar, spelling, and punctuation errors in the test set. The GEC+Fluency track added fluency errors to that list. Fluency errors include word calques, stylistically inappropriate words, repetitions, or any other constructions that sound unnatural to a native speaker. It was not mandatory to participate in both tracks, i.e., participating in either GEC-only or GEC+Fluency was acceptable.

Error classification was out of the scope of the shared task.

We provided the participants with a preprocessed version of the UA-GEC corpus (Syvokon et al., 2023) for training and validation (see Section 3 for details) but also encouraged them to use any external data of their choice. Evaluation scripts were provided together with the data.

We set up a CodaLab environment[1] to manage system submissions and the leaderboard. The participants submitted their system results to CodaLab, which automatically evaluated their results on a hidden test set and returned the scores. We used $F_{0.5}$ computed by Errant (Felice and Briscoe, 2015) as the primary metric. The leaderboard is still open for further submissions.

Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP), pages 132–137

Split	Documents	Sentences	Tokens	Annotations
Train	1,706	31,038	457,017	26,123
Valid	87	1,422	23,692	1,393
Test	79	1,274	19,911	1,081

Table 1: The GEC-only data statistics. Validation and test sets were independently annotated by two annotators.

Split	Documents	Sentences	Tokens	Annotations
Train	1,706	31,038	457,017	35,460
Valid	87	1,419	23,692	1,923
Test	79	1,271	19,911	1,423

Table 2: The GEC+Fluency data statistics. Validation and test sets were independently annotated by two annotators.

3 Data

The UNLP 2023 shared task for grammatical error correction utilizes the UA-GEC dataset (Syvokon et al., 2023) as the primary source for training, evaluation, and test data. We chose this dataset due to its relevance to the task at hand. Table 1 and Table 2 provide statistics of data used in GEC-only and GEC+Fluency tracks, respectively. The minor difference in the number of sentences is an artifact of source and target sentence alignment.

The training set comprises 1,706 documents, which amount to a total of 31,028 sentences.

For hyperparameter tuning and evaluation during development, we created a separate validation set by extracting 87 documents (1,419 sentences) from the UA-GEC test set.

In order to assess the final performance of the participating models, we formed a test set containing another 79 documents (1,271 sentences) from the remaining samples in the UA-GEC test set. Each sentence in both test set and validation set was annotated by two independent annotators. This dual annotation approach ensures a more accurate evaluation of model performance, taking into account the discrepancies and variations between human annotators.

We provide training and validation data in three formats:

- unprocessed parallel text;
- tokenized parallel text;
- .m2 files (Ng et al., 2014).

Test data is provided only as tokenized and non-tokenized source text files.

[1] https://codalab.lisn.upsaclay.fr/competitions/10740

The participants had the freedom to choose which version of the data to utilize for training their models. We employed the Stanza tokenization tool (Qi et al., 2020) to tokenize the data and prepared a tokenization script for the participants.

Preserving the document structure allowed the participants to make use of document-level context in their models. To achieve this, sentences were kept in the order in which they appeared within their respective documents. Document headers were appended before a sequence of a document's sentences to retain this structure. These headers followed a specific format: "# [0-9]{4}", where an example would be "# 1234". This approach facilitated the incorporation of document-level context while maintaining consistency across the datasets.

4 Evaluation

The primary evaluation metric used for the shared task is the $F_{0.5}$ score, which combines the precision and recall metrics while weighing precision more than recall. This metric was computed using the Errant tool (Bryant et al., 2017), a widely-accepted tool for evaluating grammatical error correction.

In addition to reporting the $F_{0.5}$ scores, the evaluation script also reports other metrics: precision, recall, true positives (TP), false positives (FP), and false negatives (FN).

Furthermore, the evaluation script reports error detection metrics. However, these are provided merely for reference and are not considered while comparing the participating models. Detection metrics can be insightful in understanding how well a system identifies errors in the text, without necessarily focusing on the correction.

All evaluation is done on tokenized data. If the participants choose to train a model that produces

Rank	Participant	TP	FP	FN	Prec	Rec	$F_{0.5}$
1	QC-NLP (fpg)	636	192	400	76.81	61.39	**73.14**
2	UA-GEC	508	**139**	496	**78.52**	50.60	70.71
3	QC-NLP (rozovska)	**661**	253	**386**	72.32	**63.13**	70.27
4	WebSpellChecker	458	170	502	72.93	47.71	65.96

Table 3: Official shared task results for all teams in Track 1. GEC-only. The best values are shown in bold.

Rank	Participant	TP	FP	FN	Prec	Rec	$F_{0.5}$
1	Pravopysnyk	580	153	742	79.13	43.87	**68.17**
2	QC-NLP (fpg)	735	269	646	73.21	53.22	68.09
3	WebSpellChecker	528	**125**	759	**80.86**	41.03	67.71
4	GrammarUA	526	138	776	79.22	40.40	66.45
5	QC-NLP (rozovska)	**739**	318	**635**	69.91	**53.78**	65.96
6	UA-GEC	594	219	745	73.06	44.36	64.69
7	Final Submission	483	212	796	69.50	37.76	59.50

Table 4: Official shared task results for all teams in Track 2. GEC+Fluency. The best values are shown in bold.

non-tokenized outputs, it must be tokenized first. We provide a tokenization script to ensure there's no mismatch in preprocessing between submission and golden data.

The train and validation sets, as well as tokenization and evaluation scripts, are published on GitHub[2].

5 Participating Systems

A total of fifteen teams registered for the UNLP 2023 shared task, but only six teams submitted their solutions before the deadline. Four teams submitted papers that were accepted to appear in the UNLP workshop proceedings and are referred to in this report. Two more teams provided their system descriptions by email.

Three teams submitted their results for both GEC-only and GEC+Fluency tracks, and three more teams submitted their results only for GEC+Fluency. We briefly review the systems here; for complete descriptions, please see the corresponding papers. Table 3 and Table 4 present the leaderboards for the two tracks.

Pravopysnyk (Bondarenko et al., 2023), the winners of the GEC+Fluency track, combined a transformer-based model with a rule-based spelling correction system. For the transformer-based model, they fine-tuned MBart (Tang et al., 2021) on UA-GEC augmented by synthetically generated errors. To generate more data, the team used round-trip translation, a custom punctuation error generation script, and replacing Ukrainian words with their Russified versions. For spelling correction, the team applied the SymSpell algorithm (Garbe, 2012) to the Ukrainian language. This algorithm uses a word frequency dictionary and a bigram frequency dictionary based on the dataset of 500k sentences collected from Ukrainian books. The most frequent word that passes the spelling criteria is then selected. The transformer-based model was responsible for most corrections. The advantages of the system include high performance, low training cost (training takes 10 minutes on Google Colab A100 GPU), and its end-to-end training setup, which allows combining different sources of synthetic data. However, the system is slower when compared to sequence tagging models.

The authors published the system on the Huggingface platform: `https://huggingface.co/Pravopysnyk/best-unlp`.

QC-NLP (Gomez et al., 2023), the winners of the GEC-only track and second place holders of the GEC+Fluency track, submitted 2 systems: (1) fpg and (2) rozovska. Both systems participated in the two tracks of the shared task. System (1) achieved stronger performance in both tracks than system (2), but system (1) requires more computational resources.

In system (1), the authors fine-tuned a pre-trained mT5-large (Rothe et al., 2021; Xue et al., 2021) to correct ungrammatical sentences to their grammatical counterparts. They first fine-tuned the model with 10M synthetically generated grammati-

[2] `https://github.com/asivokon/unlp-2023-shared-task`

cal error correction examples for three epochs and then with the shared task dataset for 10 additional epochs. The synthetic examples were generated using the approach based on the Aspell confusion sets proposed in Náplava and Straka (2019). The method was applied to the native Ukrainian data from the WNT News Crawl corpus. Fine-tuning on synthetic and learner data was done with 8 Nvidia 80GB GPUs taking approximately 16 hours to train in total.

System (2) is a transformer model proposed in Náplava and Straka (2019) pre-trained on 35M synthetic examples that use Aspell confusions and additional noise from round-trip translation and fine-tuned on the gold learner training data. Three models were trained with three different seeds, and the final model is an ensemble of the three best checkpoints. Pre-training on 1 Nvidia 32GB GPU took 7 hours per epoch for about 10 epochs until convergence. Fine-tuning took about an hour until convergence.

The authors published the systems on GitHub: https://github.com/knarfamlap/low-resource-gec-uk.

WebSpellChecker (Didenko and Sameliuk, 2023) used a custom transformer-like architecture called RedPenNet. The architecture leverages a pre-trained MLM encoder along with a shallow decoder to generate both replacement tokens and spans for editing GEC cases. During the generation of edit tokens, the encoder-decoder attention weights determine the edit spans (start and end) that point at the position of the edit in the source sentence. Edit tokens are predicted in the autoregressive way. SEP tokens separate edits in the output sequence. At each step of the feedback loop, the edit BPE token embedding is combined with a decoder-specific trainable positional encoding embedding. The resulting sum is then concatenated with the span embedding. Additionally, compact GEC task-specific decoder BPE vocabularies are trained to lower the cost of the pre-softmax dot operation, thus improving the efficiency of predicting replacement tokens.

The main advantage of RedPenNet is the ability to implement any source-to-target transformation using a minimal number of autoregressive steps, which makes it possible to effectively solve the GEC cases, including interrelated and multi-token edits. However, due to the tailored architecture of RedPenNet, there are no out-of-the-box solutions available for data preprocessing, training, or fine-tuning. Thus, convenient tools like the HuggingFace infrastructure cannot be used for rapid model fine-tuning and deployment.

The system repository: https://github.com/WebSpellChecker/unlp-2023-shared-task.

Final Submission by Maksym Tarnavskyi uses a sequence tagging GECToR model (Omelianchuk et al., 2021) that contains a transformer-based encoder stacked with two output linear layers that are responsible for error detection and error correction. The author trained the model only on UA-GEC data without any synthetic data pre-training or hyperparameter optimization. An ukr-roberta-base[3] model is used to initialize the encoder.

The system repository: https://github.com/MaksTarnavskyi/gector-large.

Model checkpoints: https://drive.google.com/drive/folders/1ZWjJwZrTQAcs48Z_h4T1Mivzf5nU3h_0.

GrammarUA by Anastasiia Hudyma uses mBART50 (Tang et al., 2020), a sequence-to-sequence model that was fine-tuned on the shared task training and validation data. This model was chosen because of good results for low-resource languages.

The author published the system on the Huggingface platform: https://huggingface.co/smartik/mbart-large-50-finetuned-gec

UA-GEC system is the baseline for Ukrainian GEC presented in Syvokon et al. (2023). The team used mBART50-large (Katsumata and Komachi, 2020; Tang et al., 2020) fine-tuned on the unprocessed training data. Training takes around 3 hours on a single Nvidia P100 GPU.

6 Conclusion

We believe that the UNLP 2023 shared task was instrumental in facilitating research on grammatical error correction for the Ukrainian language, and we hope the insights from the teams' research will be useful to the NLP community. All the data and evaluation scripts used in the shared task are available on GitHub, and the competing systems were openly published, which contributes to the reproducibility of the shared task results. The CodaLab environment remains open for further submissions, although any such submissions will be considered outside of the UNLP 2023 competition.

[3] https://huggingface.co/youscan/ukr-roberta-base

The most successful systems were submitted by Pravopysnyk (Bondarenko et al., 2023) and QC-NLP (Gomez et al., 2023), scoring 68.17% and 68.09% $F_{0.5}$ respectively on GEC+Fluency. The teams set the first state of the art results for the task of Ukrainian GEC. Notably, the common themes among the best-performing systems are fine-tuning of large pre-trained transformer-based models and synthetic data.

In the next iterations of this shared task, we plan to increase the hidden test set, include error classification, and present restricted and unrestricted tracks.

Limitations

Due to limited resources, the test set of the shared task is relatively small. More labelled data would provide for more representative results.

The $F_{0.5}$ scores in our shared task are higher when compared to similar shared tasks in other languages (Bryant et al., 2019). We attribute this to the fact that 43% of errors in the data are punctuation errors, which are easier to correct (Syvokon et al., 2023).

Breaking down system outputs by error categories would help in analyzing model performance.

Ethics Statement

Upon entering the competition, all participants of the shared task accepted the following terms and conditions of the competition:

- All participants agree to compete in a fair and honest manner in the shared task and not use any illegal, malicious, or otherwise unethical methods to gain an advantage in the shared task.
- All participants agree to not distribute or share the test data obtained during the shared task with any third parties.
- All participants agree to make their solutions publicly available upon the completion of the shared task in order to facilitate knowledge sharing and developments of the Ukrainian language.

To the best of our knowledge, the shared task participants followed these terms and conditions.

Acknowledgements

We are extremely grateful to the creators of the UA-GEC corpus who made this shared task possible. Thank you, Nastasiia Osidach, Olena Nahorna, and Pavlo Kuchmiichuk! We thank Danylo Mysak for numerous fixes and improvements of the corpus.

References

Maksym Bondarenko, Artem Yushko, Andrii Shportko, and Andrii Fedorych. 2023. Comparative study of models trained on synthetic data for Ukrainian grammatical error correction. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop*, Dubrovnik, Croatia. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.

Robert Dale, Ilya Anisimoff, and George Narroway. 2012. HOO 2012: A report on the preposition and determiner error correction shared task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 54–62, Montréal, Canada. Association for Computational Linguistics.

Robert Dale and Adam Kilgarriff. 2011. Helping our own: The HOO 2011 pilot shared task. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 242–249, Nancy, France. Association for Computational Linguistics.

Bohdan Didenko and Andrii Sameliuk. 2023. RedPenNet for grammatical error correction: Outputs to tokens, attentions to spans. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop*, Dubrovnik, Croatia. Association for Computational Linguistics.

Mariano Felice and Ted Briscoe. 2015. Towards a standard evaluation method for grammatical error detection and correction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 578–587, Denver, Colorado. Association for Computational Linguistics.

Wolf Garbe. 2012. SymSpell.

Frank Palma Gomez, Alla Rozovskaya, and Dan Roth. 2023. A low-resource approach to the grammatical error correction of Ukrainian. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop*, Dubrovnik, Croatia. Association for Computational Linguistics.

Satoru Katsumata and Mamoru Komachi. 2020. Stronger baselines for grammatical error correction using a pretrained encoder-decoder model. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 827–832, Suzhou, China. Association for Computational Linguistics.

Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2012. The effect of learner corpus size in grammatical error correction of ESL writings. In *Proceedings of COLING 2012: Posters*, pages 863–872, Mumbai, India. The COLING 2012 Organizing Committee.

Jakub Náplava and Milan Straka. 2019. Grammatical error correction in low-resource scenarios. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356, Hong Kong, China. Association for Computational Linguistics.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: A fluency corpus and benchmark for grammatical error correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vipul Raheja, and Oleksandr Skurzhanskyi. 2021. Text Simplification by Tagging. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–25, Online. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.

Oleksiy Syvokon, Olena Nahorna, Pavlo Kuchmiichuk, and Nastasiia Osidach. 2023. UA-GEC: Grammatical error correction and fluency corpus for the Ukrainian language. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop*, Dubrovnik, Croatia. Association for Computational Linguistics.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450–3466, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.