Spivavtor: An Instruction Tuned Ukrainian Text Editing Model

Aman Saini, Artem Chernodub, Vipul Raheja, Vivek Kulkarni
Grammarly
firstname.lastname@grammarly.com

We introduce Spivavtor, a dataset, and instruction-tuned models for text editing focused on the Ukrainian language. Spivavtor is the Ukrainian-focused adaptation of the English-only CoEdIT (Raheja et al., 2023) model. Similar to CoEdIT, Spivavtor performs text editing tasks by following instructions in Ukrainian like "Виправте граматику в цьому реченні" and "Спростіть це речення" which translate to "Correct the grammar in this sentence" and "Simplify this sentence" respectively. This paper describes the details of the Spivavtor-Instruct dataset and Spivavtor models. We evaluate Spivavtor on a variety of text editing tasks in Ukrainian, such as Grammatical Error Correction (GEC), Text Simplification, Coherence, and Paraphrasing, and demonstrate its superior performance on all of them. We publicly release our best-performing models and data as resources to the community to advance further research in this space.

Keywords: Ukrainian Text Editing, Instruction tuned LLMs

1. Introduction

Recently, there has been an increased focus and substantial progress in developing natural language processing (NLP) models for the Ukrainian language. These include the development of corpora like the Ukrainian Brown Corpus (Starko and Rysin, 2023), toolkits like NLP-UK[1], as well as models for word-embeddings, part-of-speech tagging, named entity recognition[2], machine translation[3], and pre-trained language models.

However, many of the aforementioned models are task-specific and do not leverage recent advances in large-scale language models and in-context learning. In particular, Large Language Models (LLMs) have demonstrated a quantum leap in generating fluent and coherent text and have thus been adapted to perform various NLP tasks (Brown et al., 2020; OpenAI, 2023; Touvron et al., 2023). Recently, LLMs have also been leveraged to perform specific text editing tasks such as GEC (Wu et al., 2023; Coyne and Sakaguchi, 2023; Fang et al., 2023) or Text Simplification (Baez and Saggion, 2023; Saggion et al., 2022). In addition to focusing on singular tasks like the above, some works have attempted to leverage LLMs and instruction fine-tuning to support multi-task text editing (Schick et al., 2023; Raheja et al., 2023; Laban et al., 2023).

There has also been extensive work on leveraging these advances to develop corresponding LLMs focused on the Ukrainian language, the most notable being UAlpaca[4], which builds a Ukrainian counterpart of the popular general-purpose instruction-tuned model – Alpaca (Taori et al., 2023). Concurrently and similarly, some research has focused on building and characterizing the capabilities of multi-lingual LLMs which are trained on massively multi-lingual data during the pre-training and instruction-tuning phases (Muennighoff et al., 2023; Workshop, 2023; Xue et al., 2021; Li et al., 2023; Wei et al., 2023; Üstün et al., 2024). While these models support instructions in Ukrainian, they do not focus on high-quality text editing tasks but on

Grammatical Error Correction
Виправте граматику в цьому реченні: А ти, батюшка, стало бути, тут в сторожі?
А ти, батюшко, стало бути, тут у сторожі?

Coherence
Покращіть зв'язність тексту: Тоді він відвідав острови Греції. Хвороба змусила його повернутися до Риму.
Потім він відвідав острови Греції, але хвороба змусила його повернутися до Риму.

Text Simplification
Спростіть речення: вода спочатку холодна, а потім волога.
Вода і холодна, і мокра.

Paraphrasing
Перефразуйте речення: Наскільки я можу судити, обидва ще живі.
Наскільки я розумію, вони обидва ще живі.

Figure 1: Example input (blue) and output (red) of the text editing tasks that Spivavtor can perform. The corresponding English translations can be found in Appendix A, Table 7.

[1] https://github.com/brown-uk/nlp_uk
[2] https://huggingface.co/lang-uk
[3] https://github.com/Helsinki-NLP/UkrainianLT
[4] https://huggingface.co/robinhad/ualpaca-7b-llama

The Third Ukrainian Natural Language Processing Workshop (UNLP) @LREC-COLING-2024, pages 95–108
25 May, 2024.

general-purpose instructions instead, such as sentiment detection, question answering, text generation, etc. However, as noted by Raheja et al. (2024), such generic instruction-tuned models are not particularly well-suited for nuanced text editing tasks without further task-specific fine-tuning. This highlights the need for an instruction-tuned model for Ukrainian that is optimized for text editing, which this paper addresses by building Spivavtor[5].

Spivavtor can follow instructions for complex text editing tasks like GEC, Text Simplification, Coherence, and Paraphrasing (Figure 1). A significant challenge to building an instruction-tuned model for Ukrainian optimized for text editing has been the limited availability of text editing datasets in Ukrainian. In this work, we address this challenge by adapting existing text editing datasets from Ukrainian and English and converting them to ”instruction-following” datasets (similar to CoEdIT and mEdIT). We then show how these newly constructed datasets can be used to build state-of-the-art text editing models for Ukrainian. Finally, through comprehensive evaluations, we empirically reveal critical insights on how the performance on Ukrainian text editing tasks is affected by various choices like model architecture, model scale, and training data mixtures. All our models and data are publicly available as resources for the community[6].

2. Related Work

Prior work falls into two major categories: (a) Ukrainian-NLP Models and (b) Multi-lingual LLMs. We discuss each of these below.

Large Language Models for Ukrainian Several works have focused on building LLMs and resources for Ukrainian. These mainly consist of manually curated Ukrainian language datasets and corpora like Starko and Rysin (2023) for Part of Speech, Syvokon et al. (2023) for Grammatical Error Correction (GEC), NER-UK for Named Entity Recognition[7], UA-SQUAD for Question Answering Ivanyuk-Skulskiy et al. (2021). Some Ukrainian datasets are also derived from large multi-lingual datasets filtered for the Ukrainian language data (for e.g., Ukrainian Tweet Corpus[8]). In addition to these datasets, custom models have also been built for the above tasks, a list of which is curated here[9]. A notable such model aimed at general instruction following in Ukrainian is the UAlpaca model, which was obtained by further fine-tuning LLaMA on Ukrainian translations of the Alpaca (Taori et al., 2023) dataset.

Text Editing via Instruction Tuning There exists extensive prior literature leveraging instruction-tuned LLMs for various text editing tasks in both monolingual and multi-lingual settings. More recently, Schick et al. (2023), Raheja et al. (2023), and Laban et al. (2023) have focused on general-purpose text editing using instruction-tuned LLMs for English. However, all of these prior approaches have been limited in monolingual settings because they focus only on English.

Text editing capabilities in the Ukrainian language have been developed only in multi-lingual settings, where most works have proposed task-specific multi-lingual models. These works have developed models for text editing tasks like GEC (Rothe et al. (2021); Sun et al. (2022)), paraphrasing (Chowdhury et al., 2022), formality style transfer (Briakou et al., 2021), and text simplification (Mallinson et al. (2020); Martin et al. (2022); Ryan et al. (2023)). However, they are similarly limited due to their singular focus on specific text editing tasks rather than high-quality, general-purpose text editing.

There exists an even more extensive literature on general-purpose multi-lingual LLMs (many of which also include support for Ukrainian (Üstün et al., 2024; Li et al., 2023)), these models generally aim for massive multi-language support and are not optimized explicitly for Ukrainian or text editing. A comprehensive review of multi-lingual LLMs is out of the scope of this paper.

Finally, our work is closest to the recently proposed mEdIT (Raheja et al., 2024), which developed a multi-lingual extension to CoEdIT with support for a similar set of tasks for six languages, but is limited in our context as it is not focused on Ukrainian as one of its core languages.

3. Spivavtor

In this section, we describe the construction of Spivavtor. Specifically, we discuss (a) Dataset construction, (b) Model architecture choices, and (c) Model training process.

3.1. Spivavtor-Instruct Dataset

Similar to prior work (Raheja et al., 2023), we consider four text editing tasks: (a) Fluency/Grammatical Error Correction (GEC),

[5] Spivavtor means “co-author” in Ukrainian.

[6] https://huggingface.co/collections/grammarly/spivavtor-660744ab14fdf5e925592dc7

[7] https://github.com/lang-uk/ner-uk

[8] https://github.com/saganoren/ukr-twi-corpus

[9] https://github.com/osyvokon/awesome-ukrainian-nlp

(b) Simplification, (c) Coherence, and (d) Paraphrasing; and construct a unified Ukrainian text editing instruction dataset which we call SPIVAVTOR-Instruct. We consider these tasks for two reasons: (a) These tasks are largely representative of the most common text editing tasks, and (b) It is feasible to obtain curated good-quality data for these tasks either in Ukrainian or English. For tasks where Ukrainian data is not readily available, we use the available English datasets to construct their Ukrainian counterpart by translating them into Ukrainian using Google Translate API[10]. Due to time constraints, we did not explore other translation services or models. Having outlined the tasks, we now discuss the task-specific datasets we used and our process for constructing SPIVAVTOR-Instruct.

GEC We use the Ukrainian Grammatical Error Correction (UA-GEC) dataset (Syvokon et al., 2023) for GEC/Fluency. This dataset contains 33k pairs of grammatically incorrect and correct sentences in Ukrainian. The original dataset contains train (31k) and test (2k) splits. However, since we explore different model choices and training hyper-parameters, we further randomly split the train set to create a custom train (28k) and validation (3k) dataset.

Simplification For the Simplification task, we adapt three English datasets: (a) WikiLarge (Zhang and Lapata, 2017), and (b) WikiAuto (Jiang et al., 2020) for training. For evaluation, we use ASSET (Alva-Manchego et al., 2020), and Turk (Xu et al., 2016a) datasets. As mentioned above, we translate all these datasets into Ukrainian using Google Cloud Translation API.

Coherence For the coherence task, which involves combining two sentences together coherently using edit operations such as inserting discourse connectives, we once again translate an English dataset, given the lack of an equivalent dataset for Ukrainian. In particular, we adapt the DiscoFuse dataset (Geva et al., 2019) and the Coherence split of ITERATER (Du et al., 2022) and translate them to Ukrainian using the Google Cloud Translation API.

Paraphrasing We adapt the popular PAWS (Zhang et al., 2019) dataset in English by constructing its Ukrainian counterpart via translation, maintaining their train and test splits. We evaluate paraphrasing on MRPC (Dolan and Brockett, 2005), STS (Cer et al., 2017), and QQP datasets.

The Ukrainian datasets we thus obtain are suitable for training Ukrainian-specific models, but they are not suitable yet for instruction tuning since they do not contain explicit instructions. To overcome this, we prepend task-specific verbalizers that describe the task to be performed as simple instructions to each instance. These task-specific verbalizers were curated by domain experts in Ukrainian. More specifically, for a given task-specific instance, we assign a specific verbalizer by randomly drawing a sample from the task-specific verbalizer set. Table 2 shows a few instruction verbalizers for each task with the full set available in Appendix Table 9. Similarly, Table 1 summarizes the number of training, validation, and test instances, along with the number of distinct instructions per task. Finally, it is to be noted that to ascertain the quality of the Ukrainian translated datasets, a random sample of 100 instances were chosen for verification by native speakers of Ukrainian and found to be largely satisfactory[11].

3.2. Models

To train SPIVAVTOR, we consider two kinds of transformer-based LLM architectures – Encoder-Decoder as well as the Decoder-only architecture. Both architectures have been shown to be generally effective in prior work (Xue et al., 2021; Üstün et al., 2024) although the Decoder-only models tend to be more popular recently with the release of models like ChatGPT and GPT4 (OpenAI, 2023). Thus, in the realm of Ukrainian text editing, we empirically explore the effect of both of these model architectures on task performance. We also explore the effect of different model sizes considering relatively smaller models with 1B parameters as well as larger models with upto 13B parameters.

3.2.1. Encoder-Decoder Models

mT5 (Xue et al., 2021) is a multi-lingual variant of T5 (Raffel et al., 2020), trained on the mC4 dataset [12], a multi-lingual variant of the C4 dataset extended to 101 languages. We experiment with two variants of mT5 – LARGE (1.2B) and XXL (13B).

mT0 (Muennighoff et al., 2023) is a family of multi-lingual Encoder-Decoder models capable of following human instructions in dozens of languages. We use the mt0-LARGE (1.2B) model. The mT0 models are constructed by fine-tuning mT5 models on the xP3 cross-lingual task mixture

[10] https://cloud.google.com/translate/docs/advanced/translating-text-v3

[11] Grossly incorrect translations were corrected manually.

[12] https://huggingface.co/datasets/mc4

Task	#Train	#Validation	#Test	#Verbalizers
GEC	27,929	3,103	2,682	9
Simplification	11,501	1,278	533	11
Coherence	9,278	1,031	551	7
Paraphrasing	14,076	1,564	6,244	13
Total	62,784	6,976	10,010	40

Table 1: Summary statistics of the Spivavtor-Instruct dataset.

Task	Verbalizers	English translation
GEC	"Виправте граматику в цьому реченні:" "Зробіть речення граматичним:" "Удосконаліть граматику цього тексту:"	"Correct the grammar in this sentence:" "Make the sentences grammatical:" "Improve the grammar of this text:"
Simplification	"Спростіть речення:" "Зробіть речення простим:" "Зробіть цей текст легше для розуміння:"	"Simplify the sentences:" "Make the sentence simple:" "Make this text easier to understand:"
Coherence	"Виправте зв'язність в реченні:" "Покращіть зв'язність тексту:" "Зробіть текст більш зв'язним:"	"Correct the coherence in the sentence:" "Improve text coherence:" "Make the text more coherent:"
Paraphrasing	"Перефразуйте речення:" "Перефразуйте цей текст:" "Напишіть перефраз для речення:"	"Rephrase the sentence:" "Paraphrase this text:" "Write a paraphrase for the sentence:"

Table 2: A subset of verbalizers for each task used as instructions in the Spivavtor-Instruct dataset (see Appendix Table 9 for full set of instructions).

dataset, which consists of multi-lingual datasets with English prompts. As a result, mT0 models are better suited for following English prompts. We also use the mt0-xxl-mt variant, which is fine-tuned on the xP3mt dataset and is better suited for prompting in non-English.

Aya 101 (Üstün et al., 2024) is a massively multilingual generative language model that follows instructions in 101 languages of which over 50% are considered low-resourced. Aya outperforms mT0 and BLOOMZ (Muennighoff et al., 2022) on the majority of tasks while covering double the number of languages. The model has 13B parameters and the same architecture as the mt5-xxl model.

3.2.2. Decoder-only LLMs

Bactrian-X (Li et al., 2023) is a collection of lightweight adapters for LLaMA (7B and 13B) (Touvron et al., 2023) and BLOOM (7B) (Workshop, 2023) on the Bactrian-X dataset, which is a multi-lingual parallel dataset containing 3.4 million instruction–response pairs across 52 languages. We use the bactrian-x-llama-7b-merged variant.

Mistral (Jiang et al., 2023) is a family of large language models. We use the Mistral-7B-Instruct-v0.2 variant which is an instruction fine-tuned version of the Mistral-7B-v0.2 model.

Llama2 Chat Models We also consider full-parameter fine-tuning of the Llama2 7B and 13B chat models. While the aforementioned Bactrian-X models also derive from the LLaMA models, they use parameter-efficient fine-tuning (PEFT), specifically, low-rank adaptation (LoRA) (Hu et al., 2022), thus, significantly reducing the number of trainable parameters during fine-tuning. Thus, in contrast to Bactrian-X models, we consider full-parameter fine-tuning of Llama-2 Chat models as well. We use the Llama-2-7b-chat-hf and Llama-2-13b-chat-hf variants.

3.3. Training

We use Spivavtor-Instruct dataset to perform instruction-tuning on both styles of models described above. We train all models using Deepspeed (Rasley et al., 2020) on 8xA100 GPU instances with AdamW optimizer, a per-device batch size of 8, and a learning rate of 5e-5. For Decoder-only models, the maximum sequence length is set to 512 tokens, whereas for Encoder-Decoder models, the maximum sequence length is set to 256 tokens for both source and target. The best-performing checkpoints were chosen based on the validation loss.

3.4. Inference

For Inference, we mostly use default generation parameters for temperature, beam size as specified in the corresponding model with the exception of max output length, which is set to the max sequence length used while training the model. To avoid repeated generation with Decoder-only models, we used the model-specific EOS tag to end decoding.

4. Evaluation

Metrics We evaluate all models on the task-specific test splits of the SPIVAVTOR-Instruct dataset. As in prior work, we report the standard evaluation metrics used for each task. In particular, we report the $F_{0.5}$ Correction score for GEC calculated using ERRANT (Bryant et al., 2017) weighing precision twice as much as recall. Following prior work by Ryan et al. (2023); Raheja et al. (2023) we report SARI (Xu et al., 2016b) for Simplification as well as Coherence. For Paraphrasing, we report BLEU (Papineni et al., 2002). In order to capture the overlap with source as well as reference, we report both reference-free BLEU (also called Self-BLEU in Zhu et al. 2018) and reference-based BLEU, since they collectively provide additional signal on paraphrasing quality than either one of them alone (see Shen et al. (2022) and Section 6).

Baselines We evaluate our SPIVAVTOR models against strong instruction tuned baseline models. In addition to the corresponding base models (i.e. not fine-tuned on SPIVAVTOR-Instruct dataset), we also evaluate against the following:

- **Copy**: The Copy baseline, which just copies the input sentence, is a surprisingly trivial but hard-to-beat baseline.
- **UAlpaca**: To ascertain the effect of task-specific instruction fine-tuning in contrast to large-scale diverse instruction fine-tuning, we consider the UAlpaca model in a zero-shot setting. UAlpaca is a LLaMA 7B model trained on Ukrainian translations of 52K diverse and generic instructions of the Alpaca dataset (Taori et al., 2023). For prompting UAlpaca, we used the recommended prompt format that it was fine-tuned on and replaced the instruction placeholder with the assigned verbalizer.
- **GPT4** Noting the widespread popularity of GPT4 (OpenAI, 2023) and a general notion that GPT4 generally obtains very strong performance on many NLP tasks, we also consider this as a baseline (in the zero-shot setting) where we prompt it with a verbalizer and the input text. In particular, we use gpt-4-0613 model with a context window of 8192 tokens and a training data cutoff of Sep 2021. To give GPT4 the best shot at success and to account for prompt sensitivity, we evaluate GPT4 on the chosen task with all possible verbalizers in our set and report the score corresponding to the best verbalizer. If there is no response received from the API due to content filtration policies, we consider the input unchanged for evaluation purposes.
- **GPT-3.5-Turbo** We also compare against the more cost effective GPT-3.5-Turbo model, widely known as ChatGPT. In particular, we use gpt-3.5-turbo version 0301.

4.1. Quantitative Results

In this section, we describe our main results and discuss findings from ablation studies to gain insights into the factors driving model performance.

Main Results Table 3 shows the performance of various models on all tasks in consideration. It presents aggregated scores for all tasks across different datasets. The dataset-specific scores for all relevant tasks are present in Appendix A, Table 8. Based on these results, we can make the following observations:

1. **SPIVAVTOR generally performs significantly better over baselines**. Comparing the performance metrics for SPIVAVTOR models to their baseline counterparts, we generally observe that SPIVAVTOR significantly outperforms baseline models (including GPT4), with Simplification being the only exception where performance is at par. This result suggests the effectiveness of domain-specific instruction tuning for superior performance on specific tasks.
2. **Domain-specific Instruction tuning outperforms instruction tuning on a large set of generic instructions.** Given the effectiveness of instruction tuning and in-context learning, a natural question arises: For text editing with instructions, is it sufficient to instruction-tune a model with a very large set of diverse instructions that are not necessarily related to text editing? We can answer this question empirically by comparing the performance of SPIVAVTOR models (that are instruction tuned on text

Model	Type	Size	GEC	Simplification	Coherence	Paraphrasing
Copy	-	-	0	21.98	26.89	100/31.4
Bactrian-X-7b	D	7B	0.65	36.76	40.37	21.86/8.13
UAlpaca-7b	D	7B	0.57	35.17	32.64	13.26/4.95
Mistral-7b	D	7B	0.3	38.96	32.41	9.30/3.79
mt0-large	ED	1.2B	0.21	29.56	22.14	6.70/2.68
aya-101	ED	13B	21.98	35.59	38.30	42.68/15.53
GPT-3.5-Turbo	D	-	1.17	40.18	44.93	26.60/12.51
GPT4	D	-	27.18	40.08	43.44	23.23/11.7
Spivavtor-Bactrian-X-7b	D	7B	55.73	36.90	47.80	65.31/23.65
Spivavtor-Mistral-7b	D	7B	51.54	34.55	44.12	76.56/25.33
Spivavtor-Llama2-7b	D	7B	55.88	36.94	48.73	48.97/18.9
Spivavtor-Llama2-13b	D	13B	56.48	36.98	48.55	57.31/21.35
Spivavtor-mt5-large	ED	1.2B	61.83	36.40	48.27	77.31/26.68
Spivavtor-mt0-large	ED	1.2B	61.44	36.16	48.28	77.83/26.73
Spivavtor-mt5-xxl	ED	13B	63.00	37.84	48.97	72.42/25.64
Spivavtor-mt0-xxl-mt	ED	13B	64.55	**38.44**	**49.48**	68.63/25.07
Spivavtor-aya-101	ED	13B	**64.57**	37.87	48.51	73.28/26.17

Table 3: Comparison of Spivavtor models against various baselines including Copy (target=source), Decoder-only(D) and Encoder-Decoder(ED) models when evaluated in a zero-shot setting. For GEC, we report **$F_{0.5}$ Correction**. For Simplification and Coherence, we report **SARI**. For Paraphrasing, we report **ref-free/ref-based BLEU** where ref-free is the reference-free BLEU and ref-based is the reference-based BLEU to capture the overlap with both source and reference. All scores have been scaled to lie between 0 and 100. Note that all Spivavtor models outperform baseline models.

Held-Out Task	GEC	Simplification	Coherence	Paraphrasing
GEC	**18.47**	37.41	52.11	71.44/26.14
Simplification	64.95	**32.84**	48.96	68.39/25.01
Coherence	62.57	36.79	**39.48**	72.86/25.81
Paraphrasing	64.25	36.86	51.84	**74.61/25.90**

Table 4: Performance of the Spivavtor-aya-101 model on all tasks when one task is ablated. We report the same metrics as in Table 3. The bolded numbers represent the zero-shot performance of the model when not trained on that particular task.

editing instructions) with UAlpaca – a model that is instruction tuned on 52K diverse instructions. From Table 3, we observe that UAlpaca has significantly lower performance compared to its equivalent Spivavtor model (Spivavtor-Llama2-7B). It may not be sufficient to instruction-tune models on just a large set of diverse instructions, and there is significant value to instruction tuning on domain-specific instructions, an observation that reaffirms findings in prior work by Raheja et al. (2023).

3. **Encoder-Decoder models outperform Decoder-only models.** Given the extensive popularity of LLMs, there has been a significant surge in the availability of LLMs. While some LLMs use an Encoder-Decoder architecture (Xue et al., 2021; Üstün et al., 2024), some others use a Decoder-only style (OpenAI, 2023; Taori et al., 2023; Touvron et al., 2023). Yet, it is not clear if one architecture offers consistently superior performance over the other and on what tasks one might prefer a specific architecture. We trained both styles of models on the Spivavtor-Instruct dataset to evaluate the results empirically. Our results indicate that Encoder-Decoder models generally outperform Decoder-only models when fine-tuned on domain-specific instructions. More specifically, note that all Spivavtor Encoder-Decoder models outperform Spivavtor Decoder-only models on average.

4. **Larger models outperform smaller ones.** Our results also suggest that, generally, larger models tend to perform better than smaller ones - both across baselines and across Spivavtor models within an architecture

family. This finding further reinforces the effectiveness of model scaling on task performance.

Task Ablation In this setting, we hold out specific tasks in a controlled manner to evaluate one of the Spivavtor models (Spivavtor-aya-101), to see how it might generalize to unseen text editing tasks. More specifically, in each turn, we hold out one of the tasks, train on the remaining set, and report task performance on all tasks. The results of this ablation study are shown in Table 4 and clearly demonstrate the usefulness of instruction tuning on all tasks. The model performs significantly better when trained on task-specific data as compared to the zero-shot setting.

4.2. Qualitative Error Analysis

In this section, we first discuss the subpar performance of most baseline models on GEC, as observed in Table 3. Careful inspection of the model outputs indicates several problems with zero-shot model evaluation. The most frequent problems include repeated generation, output generation in English instead of Ukrainian, explanation of corrections made, text generation indicating no change is needed, to name a few. These models also suffer from an overcorrecting issue (Fang et al., 2023) and tend to perform paraphrasing and fluency rewrites. As a result, in many cases, the conservative span-based $F_{0.5}$ metric (used to evaluate GEC) can't capture the correct edits, resulting in low performance.

Next, we evaluate one of our best-performing models (Spivavtor-aya-101) qualitatively. For each task, we provide the model a sample input along with an instruction on what to do and show the model-generated output for a handful of such inputs in Table 5. We also highlight some of the errors made by our model in Table 6. The English translations for all examples are provided in the same tables for reference. On the GEC task, the output quality outperforms all baseline models. Due to instruction tuning, the edits become more conservative and therefore, are better captured by $F_{0.5}$ metric using M2scorer[13]. The instruction-tuned models avoid common errors such as repetitions and generation of gibberish text and are much better at following instructions. However, the edits made are not always correct. For the simplification task, the majority of errors arise from changes in meaning due to excessive text truncation. Another typical negative pattern is the filtration of named entities and/or their replacement with pronouns. The coherence task is performed rather successfully. The model either edits the text correctly or leaves the text uncorrected. The most common issue is the incorrect usage of conjunctions, disrupting the logical flow, e.g. using "but" instead of "and", "however" instead of "so", etc. Paraphrasing is done mainly on the lexical level by changing the word or phrase order inside the text. In longer texts, such as those in the MRPC dataset, we sometimes observe a change in meaning compared to the input, whereas in shorter texts, such as those in STS and QQP, it tends to align more closely with the reference rewrites. Errors highlighting some of these problems are shown in Table 6.

[13]https://github.com/nusnlp/m2scorer

5. Conclusions

We introduce Spivavtor – an instruction-tuned LLM for Ukrainian text editing and corresponding Spivavtor-Instruct dataset. We describe in detail the construction of Spivavtor, including how we curate the instruction dataset in Ukrainian for text editing tasks. We empirically show that Spivavtor significantly outperforms other models on text editing tasks. We also analyze the effect of modeling choices (scale and architecture) on task performance. Overall, our experiments support the hypothesis that domain and task-specific instruction tuning is needed to obtain better performance on complex text editing tasks. Finally, all our datasets and models are released to the community to help advance research in the area of Ukrainian NLP.

6. Limitations

While we have introduced an instruction-tuned LLM for Ukrainian text editing, we acknowledge a few limitations of our work. First, due to the limitations of the translation API used, our training data may not be of the highest quality. This limitation could potentially be overcome by curating high-quality data from native speakers of the Ukrainian language. The scale of our training data can also be increased over our current set.

Second, while we use standard evaluation metrics for the text editing tasks, we acknowledge that many of these metrics have limitations and do not capture many aspects of text quality (e.g. meaning preservation, etc.). For example, in the case of evaluating paraphrasing, there is no single automatic metric that holistically captures all important aspects of a good paraphrase as judged by humans. In fact, even if one narrows down to using BLEU score as a metric, it has been shown in prior work that either one of reference-free BLEU score or reference-based BLEU score may correlate better with human judgments, and this may be dataset or benchmark-dependent (Shen et al., 2022) which is why we report both

GEC Input ► Виправте граматику в цьому реченні: Дякую за інформацію! ми з Надією саме вийшли з дому **Output** ► Дякую за інформацію! Ми з Надією саме вийшли з дому.	**GEC Input** ► Correct the grammar in this sentence: Thanks for the information! we with Nadia just left the house **Output** ► Thanks for the info! Nadia and I just left the house.
Simplification Input ► Спростіть речення: Там він помер через шість тижнів, 13 січня 888 року. **Output** ► Там він помер 13 січня 888 року.	**Simplification Input** ► Simplify the sentence: There he died six weeks later, on January 13, 888. **Output** ► There he died on January 13, 888.
Coherence Input ► Виправте зв'язність в реченні: Лінч досі відмовляється розповідати про сумнозвісну травневу дорожньо-транспортну пригоду, коли він збив жінку-пішохода в районі нічного клубу Баффало та відлетів. Той факт, що Лінч взагалі говорив, заслуговує на увагу в цих краях. **Output** ► Лінч досі відмовляється розповідати про сумнозвісну травневу дорожньо-транспортну пригоду, коли він збив жінку-пішохода в районі нічного клубу Баффало та відлетів. Однак той факт, що він взагалі говорив, заслуговує на увагу в цих краях.	**Coherence Input** ► Correct the coherence in the sentence: Lynch still refuses to talk about the infamous May traffic accident in which he struck a female pedestrian in a Buffalo nightclub area and drove away. The fact that Lynch spoke at all deserves attention in this place. **Output** ► Lynch still refuses to talk about the infamous May traffic accident in which he struck a female pedestrian in a Buffalo nightclub area and drove away. However, the fact that Lynch spoke at all deserves attention in this place.
Paraphrasing Input ► Перефразуйте речення: Який найкращий комплімент, який ти отримував від будь-кого? **Output** ► Який найкращий комплімент, який ти коли-небудь отримував?	**Paraphrasing Input** ► Paraphrase the sentence: What is the greatest compliment that you ever received from anyone? **Output** ► What is the greatest compliment that you ever received?

Table 5: Example inputs and outputs from Spivavtor-aya-101 model for all relevant tasks.

GEC Input ► Виправте граматичні помилки в цьому реченні: В поки що вересень будем повну оплату робити. **Output** ► У поки що вересні будем повну оплату робити.	**GEC Input** ► Correct the grammatical errors in this sentence: On the meantime in September, will we make the full payment. **Output** ► In the meantime in September, will we make the full payment.
Simplification Input ► Зробіть речення простим: Джидда є головними воротами до Мекки, найсвятішого міста ісламу, яке працездатні мусульмани повинні відвідати принаймні раз у житті. **Output** ► Це одне з головних воріт до Мекки, яке мусульмани повинні відвідати принаймні раз у житті.	**Simplification Input** ► Make the sentence simple: Jeddah is the main gateway to Mecca, Islam's holiest city, which able-bodied Muslims must visit at least once in their lifetime. **Output** ► It is one of the main gateways to Mecca that Muslims must visit at least once in their lifetime.
Coherence Input ► Виправте зв'язність в цьому тексті: Зайферт: Ця зміна здавалася певною протягом більшої частини року. Нещодавно Гуделл сказав, що очікує голосування під час зборів власників ліги в березні. **Output** ► Зайферт: Ця зміна здавалася певною протягом більшої частини року, але нещодавно Гуделл сказав, що очікує голосування під час зборів власників ліги в березні.	**Coherence Input** ► Correct the coherence in this text: Seifert: This change seemed certain for most of the year. Goodell recently said he expects a vote at the league's owners meeting in March. **Output** ► Seifert: That change seemed certain for most of the year, but Goodell recently said he expects a vote at the league's owners meeting in March.
Paraphrasing Input ► Перефразуйте це речення: Чоловік грає на музичній клавіатурі. **Output** ► Чоловік грає на клавіатурі.	**Paraphrasing Input** ► Rephrase this sentence: The man is playing the musical keyboard. **Output** ► A man plays the keyboard.

Table 6: Example errors made by Spivavtor-aya-101 model for all tasks with English translations.

reference-based and reference-free BLEU scores in our evaluations for paraphrasing. In addition to BLEU, one would also report a semantic similarity score (like BERTScore) between the paraphrase and the source to capture how semantically close the paraphrase is to the source (or reference). For English, this is typically done using popular sentence embedding models like BERT, but it is not clear what the best approach is for Ukrainian, which is why we do not consider this dimension in our evaluation. One could potentially address such limitations by directly seeking human judgments on the quality of model predictions.

Finally, while we explore different settings of hyper-parameters (like batch size and learning rate) and different variants of prompts in our experiments, our search space is not exhaustive and is limited due to computational budgets and time constraints. We also acknowledge that the performance of closed models like GPT4 may drift or change over time due to model refreshes. Even in cases where model artifacts were publicly available, one must acknowledge that they were likely pre-trained on different datasets in the pretraining stage, and the precise effect of this on our specific downstream task performance is not known and is absorbed in our model performance reports. Research around an improved characterization of such variance in expected performance would be useful in the future.

7. References

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020. ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4668–4679, Online. Association for Computational Linguistics.

Anthony Baez and Horacio Saggion. 2023. LSLlama: Fine-tuned LLaMA for lexical simplification. In *Proceedings of the Second Workshop on Text Simplification, Accessibility and Readability*, pages 102–108, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Eleftheria Briakou, Di Lu, Ke Zhang, and Joel Tetreault. 2021. Olá, bonjour, salve! XFORMAL: A benchmark for multilingual formality style transfer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3199–3216, Online. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Jishnu Ray Chowdhury, Yong Zhuang, and Shuyi Wang. 2022. Novelty controlled paraphrase generation with retrieval augmented conditional prompt tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10535–10544.

Steven Coyne and Keisuke Sakaguchi. 2023. An analysis of gpt-3's performance in grammatical error correction. *arXiv preprint arXiv:2303.14342*.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Wanyu Du, Vipul Raheja, Dhruv Kumar, Zae Myung Kim, Melissa Lopez, and Dongyeop Kang. 2022. Understanding iterative revision from human-written text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3573–3590, Dublin, Ireland. Association for Computational Linguistics.

Tao Fang, Shu Yang, Kaixin Lan, Derek F. Wong, Jinpeng Hu, Lidia S. Chao, and Yue Zhang. 2023. Is chatgpt a highly fluent grammatical error correction system? a comprehensive evaluation.

Mor Geva, Eric Malmi, Idan Szpektor, and Jonathan Berant. 2019. DiscoFuse: A large-scale dataset for discourse-based sentence fusion. 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 3443–3455, Minneapolis, Minnesota. Association for Computational Linguistics.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Bogdan Ivanyuk-Skulskiy, Anton Zaliznyi, Oleksand Reshetar, Oleksiy Protsyk, Bohdan Romanchuk, and Vladyslav Shpihanovych. 2021. uadatasets: a collection of ukrainian language datasets.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural CRF model for sentence alignment in text simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7943–7960, Online. Association for Computational Linguistics.

Philippe Laban, Jesse Vig, Marti A Hearst, Caiming Xiong, and Chien-Sheng Wu. 2023. Beyond the chat: Executable and verifiable text-editing with llms. *arXiv preprint arXiv:2309.15337*.

Haonan Li, Fajri Koto, Minghao Wu, Alham Fikri Aji, and Timothy Baldwin. 2023. Bactrian-x : A multilingual replicable instruction-following model with low-rank adaptation.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2020. Zero-shot crosslingual sentence simplification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5109–5126, Online. Association for Computational Linguistics.

Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot. 2022. MUSS: Multilingual unsupervised sentence simplification by mining paraphrases. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1651–1664, Marseille, France. European Language Resources Association.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Vipul Raheja, Dimitris Alikaniotis, Vivek Kulkarni, Bashar Alhafni, and Dhruv Kumar. 2024. medit: Multilingual text editing via instruction tuning.

Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. 2023. CoEdIT: Text editing by task-specific instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5274–5291, Singapore. Association for Computational Linguistics.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3505–3506, New York, NY, USA. Association for Computing Machinery.

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.

Michael Ryan, Tarek Naous, and Wei Xu. 2023. Revisiting non-English text simplification: A unified multilingual benchmark. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4898–4927, Toronto, Canada. Association for Computational Linguistics.

Horacio Saggion, Sanja Štajner, Daniel Ferrés, Kim Cheng Sheang, Matthew Shardlow, Kai North, and Marcos Zampieri. 2022. Findings of the TSAR-2022 shared task on multilingual lexical simplification. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 271–283, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Timo Schick, Jane A. Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. 2023. PEER: A collaborative language model. In *The Eleventh International Conference on Learning Representations*.

Lingfeng Shen, Lemao Liu, Haiyun Jiang, and Shuming Shi. 2022. On the evaluation metrics for paraphrase generation. *arXiv preprint arXiv:2202.08479*.

Vasyl Starko and Andriy Rysin. 2023. Creating a POS gold standard corpus of Modern Ukrainian. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP)*, pages 91–95, Dubrovnik, Croatia. Association for Computational Linguistics.

Xin Sun, Tao Ge, Shuming Ma, Jingjing Li, Furu Wei, and Houfeng Wang. 2022. A unified strategy for multilingual grammatical error correction with pre-trained cross-lingual language model.

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 (UNLP)*, pages 96–102, Dubrovnik, Croatia. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Xiangpeng Wei, Haoran Wei, Huan Lin, Tianhao Li, Pei Zhang, Xingzhang Ren, Mei Li, Yu Wan, Zhiwei Cao, Binbin Xie, Tianxiang Hu, Shangjie Li, Binyuan Hui, Bowen Yu, Dayiheng Liu, Baosong Yang, Fei Huang, and Jun Xie. 2023. Polylm: An open source polyglot large language model.

BigScience Workshop. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Haoran Wu, Wenxuan Wang, Yuxuan Wan, Wenxiang Jiao, and Michael Lyu. 2023. Chatgpt or grammarly? evaluating chatgpt on grammatical error correction benchmark.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016a. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016b. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

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.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594, Copenhagen, Denmark. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models.

Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model. *arXiv preprint arXiv:2402.07827*.

8. Appendix A

GEC Input ▶ Виправте граматику в цьому реченні: А ти, батюшка, стало бути, тут в сторожі? **Output** ▶ А ти, батюшко, стало бути, тут у сторожі?	**GEC Input** ▶ Correct the grammar in this sentence: And you, father, are you here in guard duty? **Output** ▶ And you, father, are you here on guard duty?
Coherence Input ▶ Покращіть зв'язність тексту: Тоді він відвідав острови Греції. Хвороба змусила його повернутися до Риму. **Output** ▶ Потім він відвідав острови Греції, але хвороба змусила його повернутися до Риму.	**Coherence Input** ▶ Improve the coherence of the text: Then he visited the islands of Greece. Illness forced him to return to Rome. **Output** ▶ He then visited the islands of Greece, but illness forced him to return to Rome.
Simplification Input ▶ Спростіть речення: вода спочатку холодна, а потім волога. **Output** ▶ Вода і холодна, і мокра.	**Simplification Input** ▶ Simplify the sentence: first the water is cold, and then it is wet. **Output** ▶ The water is both cold and wet.
Paraphrasing Input ▶ Перефразуйте речення: Наскільки я можу судити, обидва ще живі. **Output** ▶ Наскільки я розумію, вони обидва ще живі.	**Paraphrasing Input** ▶ Rephrase the sentence: As far as I can tell, both are still alive. **Output** ▶ As far as I understand, they are both still alive.

Table 7: Example model inputs and outputs of the text editing tasks that Spivavtor can perform. English translations of the examples in Figure 1 are provided for reference.

Model	**Text Editing Tasks**						
	Simplification		**Coherence**		**Paraphrasing**		
	Asset	Turk	Sports	Wiki	MRPC	STS	QQP
Copy	17.75	24.04	26.61	28.37	100/39.90	100/38.80	100/26.20
Bactrian-X-7b	36.02	37.13	40.7	38.62	65.5/29.20	45.6/20.4	13.5/4
UAlpaca-7b	33.54	35.96	32.48	33.45	57.6/24.2	20.5/9.6	6.2/1.8
Mistral-7b	39.85	38.54	32.75	30.58	37.6/18	16.9/8.3	5.9/2
mt0-large	32.91	27.94	22.32	21.20	10.8/5.2	4.7/2.3	4.7/1.3
aya-101	32.02	37.32	38.42	37.68	78.5/34.5	56.8/25	32.1/9.8
GPT-3.5-Turbo	42.52	39.04	44.84	45.44	33.2/17.6	24/12.4	22.9/9
GPT4	42.20	39.05	43.35	43.96	29.2/16.1	17/12.7	19.9/9
Spivavtor-Bactrian-X-7b	35.15	37.75	47.29	50.53	63.2/29.1	67.1/31.9	66.5/20.2
Spivavtor-Mistral-7b	31.73	35.92	44.01	44.72	75.1/31.8	81/31.7	77.3/21.3
Spivavtor-Llama-2-7b-chat	39.29	35.80	48.13	51.95	46.2/22.3	50.7/22.3	50.5/16.7
Spivavtor-Llama-2-13b-chat	37.09	36.93	47.54	53.94	55.5/26.6	57.9/24.6	58.3/18.1
Spivavtor-mt5-large	34.82	37.17	47.97	49.87	71/32	78/34.8	80.7/23.3
Spivavtor-mt0-large	33.85	37.28	48.25	48.41	71.5/32.3	79.4/34.3	81.2/23.2
Spivavtor-mt5-xxl	38.50	37.52	48.87	49.53	67.3/30.9	69.1/30.5	75.3/22.3
Spivavtor-mt0-xxl-mt	38.95	38.20	48.67	53.80	65.4/30.4	69.6/34.8	70.4/21.6
Spivavtor-aya-101	37.71	37.95	47.87	51.94	69.9/31.6	71.7/33.3	74.2/22.5

Table 8: Comparison of Spivavtor models against various baselines, categorized by constituent datasets. We report detailed metrics for each dataset within a task. GEC is not relevant here since it is a single dataset. For Simplification and Coherence, we report SARI. For Paraphrasing, we report reference-free / reference-based BLEU just as in Table 3. All scores have been scaled to lie between 0 and 100.

Task	Verbalizers	English translation
GEC	“Виправте граматику в цьому реченні:”	“Correct the grammar in this sentence:”
	“Виправте граматичні помилки в цьому реченні:”	“Correct the grammatical errors in this sentence:”
	“Удосконаліть граматику цього тексту:”	“Improve the grammar of this text:”
	“Виправте всі граматичні помилки:”	“Correct all grammatical errors:”
	“Зробіть речення граматичним:”	“Make the sentence grammatical:”
	“Видаліть граматичні помилки:”	“Remove grammatical errors:”
	“Виправте помилки в цьому тексті:”	“Correct the errors in this text:”
	“Виправте граматичні помилки:”	“Correct the grammatical errors:”
	“Виправити граматику:”	“Correct the grammar:”
Simplification	“Спростіть речення:”	“Simplify the sentences:”
	“Напишіть простішу версію для речення:”	“Write a simpler version for the sentence:”
	“Спростіть це речення:”	“Simplify this sentence:”
	“Зробіть речення простим:”	“Make the sentence simple:”
	“Спростіть цей текст:”	“Simplify this text:”
	“Перепишіть речення так, щоб воно було простішим:”	“Rewrite the sentence so that it is simpler:”
	“Перепишіть це речення простіше:”	“Rewrite this sentence more simply:”
	“Зробіть речення простіше:”	“Make the sentences simpler:”
	“Спростіть цей текст:”	“Simplify this text:”
	“Використовуйте простіші слова:”	“Use simpler words:”
	“Зробіть цей текст легше для розуміння:”	“Make this text easier to understand:”
Coherence	“Виправте зв’язність в реченні:”	“Correct the coherence in the sentence:”
	“Покращіть зв’язність тексту:”	“Improve text coherence:”
	“Виправте зв’язність в цьому тексті:”	“Correct the coherence in this text.”
	“Виправте відсутність зв’язності в реченні:”	“Correct the lack of coherence in the sentence:”
	“Виправте зв’язність в тексті:”	“Correct the coherence in the text:”
	“Виправте зв’язність речення:”	“Correct the coherence of the sentence:”
	“Зробіть текст більш зв’язним:”	“Make the text more coherent:”
Paraphrasing	“Перефразуйте речення:”	“Rephrase the sentence:”
	“Перепишіть речення іншими словами:”	“Rewrite the sentence in other words:”
	“Перефразуйте цей текст:”	“Paraphrase this text:”
	“Перефразуйте це речення:”	“Rephrase this sentence:”
	“Перефразуйте:”	“Paraphrase:”
	“Напишіть перефраз для речення:”	“Write a paraphrase for the sentence:”
	“Напишіть перефразовану версію речення:”	“Write a paraphrased version of the sentence:”
	“Перепишіть це речення:”	“Rewrite this sentence:”
	“Перепишіть цей текст:”	“Rewrite this text:”
	“Переформулюйте це речення:”	“Rephrase this sentence:”
	“Перефразуйте це речення:”	“Paraphrase this sentence.”
	“Переформулюйте цей текст:”	“Rephrase this text:”

Table 9: A complete list of verbalizers for each task used as instructions in the SPIVAVTOR-Instruct dataset. The English translations are provided for reference.