Russian Paraphrasers: Paraphrase with Transformers

Alena Fenogenova
SberDevices, Sberbank, Russia
alenush93@gmail.com

Abstract

This paper focuses on generation methods for paraphrasing in the Russian language. There are several transformer-based models (Russian and multilingual) trained on a collected corpus of paraphrases. We compare different models, contrast the quality of paraphrases using different ranking methods and apply paraphrasing methods in the context of augmentation procedure for different tasks. The contributions of the work are the combined paraphrasing dataset, fine-tuned generated models for Russian paraphrasing task and additionally the open source tool for simple usage of the paraphrasers.

1 Introduction

One of the prominent features of any natural language is its diversity. Variability and ambiguity of natural languages lead to infinity of sequence combinations and one can always form a new sentence that has never been said before. However, there are approaches to automatic construction of texts with roughly the same meaning: paraphrases. Paraphrasing is expressing the meaning of an input sequence in alternative ways while maintaining grammatical and syntactical correctness. Paraphrases are of a great use in diverse applications on downstream NLP tasks and are presented in two main task forms: 1) Paraphrase identification - detecting if a pair of text inputs has the same meaning; classification task. 2) Paraphrase generation - producing paraphrases allows for the creation of more varied and fluent text; generation task.

The identification of paraphrases is very useful in many tasks, such as multi-document summarization (identifying paraphrases allows to condense information repeated across documents), question answering (checking the sequences of the tests, keyword matching to find answers), semantic parsing and search (to find the same queries or documents) and many others (Lewis et al., 2020).

In this work we will discuss paraphrase generation applicability. Paraphrase generation is used in different NLP applications (for example, in chatbots to diversify responses (Lippe et al., 2020)) and sub-tasks. Paraphrasers can be used to augment datasets with new data. For question answering systems, paraphrasing questions can not just increase the number of data examples for training ML-models (Xu et al., 2020), but are also used to match them with key words in the knowledge base. Paraphrasers can help generate adversarial examples to evaluate model robustness - increasing the stability of ML-models: training models on a wide variety of examples in different styles, with different sentiment, but the same meaning or intent of the user. The demand for targeting paraphrasers for generating specific writing styles is also trending now (Xu et al., 2012; Bolshakov and Gelbukh, 2004). This type of paraphrasing performs different types of style transfer, such as changing style from rude to polite, or from professional to simple language.

There are some general approaches for paraphrase generation. Rule-based approaches (Meteer and Shaked, 1988) and data-driven methods (Madnani and Dorr, 2010) are the oldest ones. Currently, the most common approach is to consider the task as supervised learning using sequence-to-sequence models (Gupta et al., 2018). The unsupervised approaches (Niu et al., 2020) are also very common. Other methods proposed include use of Deep Reinforcement Learning (Qian et al., 2019; Siddique et al., 2020). Fine-tuning with large language models such as GPT2 is also a valuable approach that can be considered supervised (Witteveen and Andrews, 2019) or unsupervised (Hegde and Patil, 2020).

The majority of the resources and methods for

Proceedings of the 8th BSNLP Workshop on Balto-Slavic Natural Language Processing, pages 11–19

paraphrasing are proposed for the English language. For the Russian language there were several attempts of paraphrase corpora creation (Pronoza et al., 2015; Gudkov et al., 2020). In 2016 the collection of the Russian paraphrase corpus and the Paraphrase Detection Shared Task (Pivovarova et al., 2017) were organized, which attracted attention to the topic and led to a number of further works on the identification of paraphrases (Kuratov and Arkhipov, 2019) and sentence similarity experiments (Kravchenko, 2017; Boyarsky and Kanevsky, 2017).

In this paper, we compare different language models for paraphrase generation in Russian, namely rugpt2-large, rugpt3-large, and multilingual models - mT5. We prove that all these models can generate good Russian paraphrases and test different ranking methods on generated examples. We provide the combined paraphrasing dataset, fine-tuned generated models for Russian paraphrasing task, augmentation experiments on data for common NLP tasks, and additionally present the open source tool for user-friendly usage of the Russian paraphrasers.

This paper is structured as follows: in section 2 we present the methodology - the dataset we use 2.1, models we fine-tune 2.2, and range strategies for paraphrasers output 2.3; section 3 is devoted to evaluation and analysis of the paraphraser performance and results - the models scores 3.1, the augmentation procedure with paraphrasers 3.2, and 3.3 the discussion about the results in the context of paraphrase application; and section 4 concludes the paper.

2 Methodology

Language models achieve impressive results when trained in a semi-supervised manner to predict the next word or words in a sequence. They can be fine-tuned and used for a variety of downstream NLP tasks (text classification, sentiment analysis, NER etc.). Good examples of such large language models that can be used for text generation are GPT-2, GPT-3, and mT5. In this section, we present our experiments with these models for Russian trained on the prepared dataset.

2.1 Dataset

Historically there are several approaches that have been used to construct paraphrasing datasets.

1. *translation-based paraphrasing* is based on the parallel data from different languages - if two Russian texts are translated to the same text in another language, then they are likely paraphrases of each other;

2. *argument-distribution paraphrasing* - if two predicates have the same meaning and they normally appear with the same arguments, they could be changed with their vector pairs;

3. *event-based paraphrasing* - the source for paraphrases is multiple descriptions of the same news event, as various news reporters are likely to choose different words to describe the same event.

Dataset	Total	news	speech
Train	410k	210k	200k
Validation	200k	100k	100k
Test	4017	2056	1961

Table 1: The dataset statistics and distribution.

The event-based approach was chosen for the creation of the Russian paraphrase corpus (Pivovarova et al., 2017). For experiments in this paper the dataset we use consists of two main parts: 1) news data from ParaPhraserPlus[1] for train and validation set and Shared task golden test for test set 2) conversational data from subtitles[2] (that were generated in an argument-distribution approach) and dialogues of users with chatbots (further in the text called *speech*). The distribution of the parts and data sizes are presented in Table 1. The test set was checked manually and further in the evaluation we assume that golden set contains high quality paraphrases to compare with.

Thus, the dataset presents two domains: informal style (speech subset, also presented in question form) and formal (news headlines). The speech subset of the data was checked for grammatical errors and typos with Yandex.Speller[3]. It was also filtered by metrics ROUGE-L (Lin, 2004) with threshold between 0.95 and 0.5. The example of the data is presented in Figure 1.

The news subset of the corpus was converted into the format of sentence pairs: $sentence_i == sentenceparaphrase_i$. Additionally, we automatically checked the cases when the information in

[1] http://paraphraser.ru/download/
[2] https://github.com/rysshe/paraphrase/tree/master/data
[3] https://yandex.ru/dev/speller/

the paraphrase sentence was excessive, for instance, $sentence_1$ *Jose Mourinho on the verge of being fired at Manchester United.* and $sentence_2$ *Mourinho could be fired if Manchester United lose to Burnley on Saturday*. The second sentence contains more information about the game, it is timing and the opposing team; in data it is permissible to have extra information in the reference sentence, but not in the paraphrase. Our experiments show that the generative models (fine-tuned on such structured data) generated more diverse sentences with absolutely out of control new information and names that could not be defined as paraphrases. It was the reason for the filtration of the pairs, where paraphrase sentence has length much longer than reference sentence or contains significantly more NER, date, and address information (the tool natasha[4] was used to detect entities). We set thresholds empirically and not strictly in order to exclude extremely inappropriate cases and kept the sentences where the entities or their number are the same, such as, *Poroshenko asked **the head of Turkey** not to recognize the presidential elections in the Crimea* and *Poroshenko urged **Erdogan** not to recognize the presidential elections in Crimea.*

2.2 Models

The idea of paraphrase generation is to build a model that reads a sequence of words and then generates a different sequence of words with the same meaning. Paraphrase generation task can be defined as generating a target sentence T for a reference sentence P where the newly generated target sentence T is semantically similar to reference sentence P.

We chose three pre-trained language models that are available for Russian:

1. ruGPT2-large[5] is a model by SberDevices team trained as a Russian analogue of OpenAI GPT-2 model (Radford et al., 2019). GPT-2 is an auto-regressive model, has up-to 1.5 Billion parameters, was trained on 40GB of Internet text to predict the next word. ruGPT2 was trained on 1024 context length with transformers on 170GB data on 64 GPUs 3 weeks.

2. ruGPT3-large is almost analogous to famous GPT-3 (Brown et al., 2020). ruGPT3 was trained on Internet text on 1024 context length with transformers on 80 billion tokens around 3 epochs, and then was fine-tuned on 2048 context.

3. mT5 (Xue et al., 2020) - Multilingual T5 (mT5) by Google is a massively multilingual pre-trained text-to-text transformer model, trained on the mC4 corpus, covering 101 languages including Russian. We trained three mT5 models on the same data: mT5-small, mT5-base and mT5-large.

We used Huggingface Transformers Library[6] to fine-tune the models on a sentence reconstruction task to generate paraphrases. Input data for GPT-based models were in the format:

$$< s > P_i === T_i < /s >$$

. Input data for mT5 models contained the sequence "rephrase: " and looked like the following:

$$perephrasiruj : P_i < /s >$$

and target format:

$$< s > T_i < /s >$$

.

All the models were trained on a single GPU Tesla V100-SXM3 32 Gb for 3-5 epochs takes 28 minutes per epoch; validation set's perplexity was used to do early stopping.

Once the model was fine-tuned, it was able to produce paraphrases in a way it was trained. If one fed in any reference phrase with the same sequence token "===" or "rephrase:", the model generated paraphrases on demand.

2.3 Candidate range

After the model was trained, we sampled from the model from test sentences as conditional input. It allowed to generate different multiple candidate sentences for the single reference sentence. We have tested different parameters (we use the interface of Hugging face, so the parameters are basic for generation: top_p, top_k sampling parameters, $temperature$, etc.), but finally used $temperature = 1.0$, $top_k = 10$, $top_p = 0.9$, $maxlength = length(P) + 10$, $repetitionpenalty = 1.5$ for GPT-based models

[4]https://github.com/natasha/natasha

[5]https://github.com/sberbank-ai/ru-gpts

[6]https://huggingface.co/

```
0.749 Куда отправиться на каникулах? => Куда поехать на каникулах?
0.799 Куда бы тебе хотелось поехать? => Куда бы тебе хотелось отправиться?
0.714 В какую страну тебе хотелось бы направиться? => В какое государство тебе хотелось бы направиться?
0.727 Куда тебе хотелось бы отправиться сейчас? => Куда тебе хотелось бы съездить?
0.545 Можешь назвать любимое мобильное приложение? => Какое твоё самое любимое мобильное приложение?
0.666 Твоё самое любимое моб приложение? => Есть любимое моб приложение?
0.545 Какое у тебя самое любимое приложение?  => У тебя какое любимое приложение?
```

Figure 1: Dialogues data example

and $temperature = 1.0$, $top_k = 50$, $top_p = 0.95$, $maxlength = 150$, $repetitionpenalty = 1.5$ - for mT5 models.

Still the quality of generating multiple outputs varies, and we can select from n examples (where $n = 10$) the best quality paraphrases based on a number of criteria or one of the range strategies to filter output down to a set of satisfactory results.

We suggest 5 types of the candidate range: 1) cosine sentence similarity between reference sentence and generated one; 2) pairwise cosine sentence similarity between n generated sentences and the reference; 3) syntax based approach; 4) BLEU best; 5) ROUGE-L best.

The first two strategies are based on sentence similarity scores received with SentenceTransformers[7]. It is a Python framework for state-of-the-art sentence and text embeddings, created based on the initial paper Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks (Reimers and Gurevych, 2019). One can use this framework to compute sentence / text embeddings for more than 100 languages. These embeddings can then be compared with cosine-similarity e.g. to find sentences with a similar meaning. We used *paraphrase-xlm-r-multilingual-v1* (Reimers and Gurevych, 2020; Thakur et al., 2020) model for paraphrase identification task (paraphrase mining). It is a multilingual version (including Russian) of *distilroberta-base-paraphrase-v1* (multilingual knowledge distilled version of multilingual Universal Sentence Encoder), trained on parallel data for 50+ languages. In the first strategy we ranged pairwise n candidates comparing cosine sentence similarities between a reference sentence and the generated one and chose the best ones (or set a distance threshold from which we are confident it is a good paraphrase).

In the second strategy we used paraphrase mining[8], the task of finding paraphrases in a corpus of sentences. The framework allows to find paraphrases in a list of sentences. Thus, we input all generated sentences and the reference one and model outputs the paraphrases.

The syntax based approach is based on the idea that arguments in the reference sentences and in the target sentence will be the same. Thus we can count the number of syntactic subjects and repeated tokens in both sentences and range the most coincidental ones. For syntax parsing we used Deeppavlov Bert Syntargus model[9]. The final two range strategies were using ROUGE-L scores and BLEU pairwise scores to choose the best from n candidates. We eliminated candidates with scores more than 0.9 and less than 0.3.

The most stable range strategy in our experiments was the first one - cosine sentence similarity between a reference sentence and the generated candidate.

3 Evaluation

We propose a two-step evaluation procedure: 1) universal metrics between gold testset examples and generated models outputs and 2) application of paraphrasers on downstream tasks where we augment data. Additionally, we will discuss the quality of the paraphrases evaluated by humans on subset of examples.

3.1 Results

To measure the quality of paraphrase generation we used average ROUGE-L, BLEU-n metrics and average cosine distance between reference and generated (calculated with model for paraphrase

[7]https://github.com/UKPLab/sentence-transformers

[8]https://www.sbert.net/examples/applications/paraphrase-mining/README.html

[9]http://docs.deeppavlov.ai/en/master/features/models/syntaxparser.html

identification task) sentences. BLEU-n calculates n gram overlap (unigrams, bigrams and trigrams), ROUGE-L measures the longest matching sequence.

Thus, we first ranged candidates as described in section 2.3, counted average scores between them for each example in the testset and got average scores for all testset examples. The results of the models are presented in Table 3. It is worth mentioning that in the process of ranging candidates we eliminated examples that were very similar by Levenshtein distance with the reference sentence (the cases when paraphraser changes case or adds punctuation symbols are not what we want).

We can observe that the golden set results have the highest scores, still the average results of the filtered models are high. The best results by ROUGE and BLEU scores are demonstrated by the mT5-small model, however it is interesting that the mt5-base and large models scores are lower, while the average candidates cosine similarity in these models is higher. It is due to the fact that if we explore generated sequences we find out that the mT5 model generates sequences that do not have great variability. For example, it is likely to generate sentences that differ only in punctuation symbols or prepositions from the reference sentence. In other words, the metrics of average cosine similarity is more reliable when paraphrases are expected to be more diverse. Thus, in order to choose the best model one need to pay attention to the metrics which are more appropriate for one's task.

The range step of the candidates is of a great importance. We took the results of mT5-small and gpt3 models. In Table 2 we present the scores depending on the different range strategies. One can see that the results vary a lot. Without filtration GPT-based model performed much worse by all the metrics. The mT5 model after filtration had even higher scores by BLEU and ROUGE metrics, but they decreased with average cosine similarity. Therefore, depending on the model and the result one expects, the range strategy should be different.

3.2 Data augmentation

In addition to general metrics, we tested if augmenting the training data with the use of paraphrasers could help to improve the performance of the model on the down-stream tasks. For this purpose we applied fine-tuned models to paraphrase examples in the training samples and, thus, augmenting the training data.

To demonstrate how paraphrases perform with default parameters on a down-stream task, we chose the following datasets:

1. RuSentiment[10] (Rogers et al., 2018) - dataset for sentiment analysis of social media posts in Russian.

2. TERRa (Shavrina et al., 2020) - Textual Entailment Recognition for Russian, a part of Russian SuperGLUE benchmark[11]. This task requires to recognize, given two text fragments, whether the meaning of one text is entailed (can be inferred) from the other text. To augment data we paraphrased the premise in each sample, kept the hypothesis and the labels, but shuffled the extended training set.

3. DaNetQA (Glushkova et al., 2020) - Russian yes/no Question Answering Dataset, a part of Russian SuperGLUE benchmark (Shavrina et al., 2020). In this dataset we paraphrased only questions and kept the paragraphs in the original format.

We took mT5-base model with range strategy of pairwise cosine similarity and default parameters. For each task we created a baseline solution as an example of paraphrasers applicability on simple setups for common tasks. For DaNetQA we made simple sequence classification with *DeepPavlov/rubert-base-cased* embeddings trained 10 epochs. For ruSentiment we used a Logistic regression classifier as a baseline. The tf-idf baseline provided by the organizers was used for TERRa.

We can see in Table 4 that the results are slightly different for all the tasks. On the TERRa task there was an increase in the performance. However, in ruSentiment we observed decrease of performance on the test set, as well as in DaNetQA, where the quality was almost the same. During the evaluation procedure the performance on training set for all the tasks was increasing. It is worth to mention that we use paraphrasers from the library with the default, same parameters for all three tasks, and even with them the results do not decrease significantly.

The results of the experiment are quite controversial. On the one hand, we did not observe a significant decrease in the performance on the set,

[10] http://text-machine.cs.uml.edu/projects/rusentiment/
[11] https://russiansuperglue.com/tasks

Model	Strategy	CS	BLEU-1	BLEU-2	BLEU-3	ROUGE-L
mT5-small	cosine similarity	0.781	0.49	0.35	0.21	0.49
mT5-small	paraphrase mining	0.776	0.58	0.45	0.30	0.58
mT5-small	syntax	0.775	0.55	0.41	0.26	0.56
mT5-small	best rouge	0.770	0.44	0.29	0.15	0.43
mT5-small	best bleu	0.772	0.53	0.40	0.26	0.53
mT5-small	all candidates	0.761	0.54	0.40	0.27	0.54
rugpt3	cosine similarity	0.754	0.41	0.27	0.15	0.42
rugpt3	paraphrase mining	0.740	0.42	0.28	0.17	0.43
rugpt3	syntax	0.737	0.42	0.27	0.16	0.42
rugpt3	best rouge	0.733	0.35	0.22	0.12	0.37
rugpt3	best bleu	0.735	0.37	0.24	0.13	0.38
rugpt3	all candidates	0.727	0.36	0.24	0.13	0.38

Table 2: Scores of the mT5-small model and rugpt3 with different range strategies.

Paraphraser	Cosine similarity	BLEU-1	BLEU-2	BLEU-3	ROUGE-L
golden set	0.848	0.57	0.43	0.28	0.58
mT5-small	0.781	0.49	0.35	0.21	0.49
mT5-base	0.798	0.35	0.23	0.12	0.37
mT5-large	0.802	0.40	0.25	0.12	0.41
rugpt2	0.717	0.43	0.29	0.17	0.44
rugpt3	0.754	0.41	0.27	0.15	0.42

Table 3: Scores of the models average

Dataset	Orig	Aug	+examples
DaNetQA	0.621	0.62	1750
ruSent	0.674	0.666	5550
TERRa	0.471	0.475	5600

Table 4: Augmentation results on test sets of the mT5-base model and number of generated examples that were added.

on the other hand, we suppose to see increase of performance with larger sizes of the dataset. One of possible explanations for this is that there was no new information in the added training set, the labels and the meaning were the same, which caused better performance during the training stage and possible overfitting. Examples of sentiment data are very short, with specific lexicon, emojis etc., which also could influence the results. DaNetQA dataset assumes YES/NO questions format, while the paraphraser could change the form of the question heavily and decrease the performance. Additionally, we believe that for every downstream task it is essential to choose a model and parameters more appropriate for the data on each step: generation, ranking, and evaluation. However, these hypotheses need further augmentation testing.

3.3 Discussion

The generated sentences are of different quality; all of the fine-tuned models are able to produce appropriate paraphrases, as well as some of them contain extra information, some typos, agreement errors or different meaning. In Figure 2 one can see the best candidates examples for three of the models. Table 5 represents the distribution of three classes: 1) good, 2) bad and 3) paraphrasers with extra information or some grammatical errors. We took 50 examples from testset, generated the paraphrases with each model, took best candidates and manually checked the number of examples for each class. The GPT2 is more stable; GPT3 is tend to produce more diverse paraphrases and add extra information that changes sense or makes it controversial; mT5 model makes more mistakes or instead changes the reference sentence not much.

The number of inappropriate candidates is significant and the procedure of candidates range and model parameters setup is crucial and should be specific for every task where we want to use paraphrasers generated on large language models. Each model has its own generating traits. For instance, GPT-based models are likely to generate

more off-top sentences and the diversity of their answers is high. We also noticed the tendency of GPT-based models to change the quality of generation examples depending on the max length. mT5 models prefer to change sentences in small pieces: change argument in the sentence on its synonym or add/cut more punctuation and symbols. Therefore, mT5-base results are rated higher with BLEU and ROUGE scores, but the examples do not differ much from reference sentences. The suitability of multilingual models to Russian has no doubts, the results are comparable. Additionally mT5-models are much faster in generation than GPT-based.

We believe the paraphrasers will be useful in many applications, thus we provide the dataset and fine-tuned models in hugging-face format in open source. The library with paraphrasers and some of range strategies for them is also available[12]. We hope everyone can find the perfect Russian paraphraser for oneself.

Model	Good	Extra info/Typos	Bad
GPT2	70%	17%	13%
GPT3	56%	34%	10%
mT5-base	63%	21%	16%

Table 5: Human evaluation of paraphrasers performance. All results were scored manually by people. The distribution is presented in percentage.

4 Conclusion

Paraphrase generation with large language models achieves impressive results. Our experiments show that both multilingual and Russian-oriented models are able to quickly learn the task of paraphrasing through fine-tuning training on a prepared Russian set of paraphrase examples. This paper contributions are the corpus of paraphrases, 5 fine-tuned models for the Russian language, comparison of them, range strategies for finding the best candidates, and the open source library in Python for convenient use of the pre-trained paraphrasers.

In future work, we would like to further explore the effectiveness of generated paraphrasers for different augmentation experiments and evaluate the models robustness in terms of reconstruction and generated paraphrases.

[12]https://github.com/RussianNLP/russian$_paraphrasers$

References

Igor A Bolshakov and Alexander Gelbukh. 2004. Synonymous paraphrasing using wordnet and internet. In *International Conference on Application of Natural Language to Information Systems*, pages 312–323. Springer.

Kirill Boyarsky and Eugeni Kanevsky. 2017. Effect of semantic parsing depth on the identification of paraphrases in russian texts. In *Conference on Artificial Intelligence and Natural Language*, pages 226–241. Springer.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Taisia Glushkova, Alexey Machnev, Alena Fenogenova, Tatiana Shavrina, Ekaterina Artemova, and Dmitry I Ignatov. 2020. Danetqa: a yes/no question answering dataset for the russian language. *arXiv preprint arXiv:2010.02605*.

Vadim Gudkov, Olga Mitrofanova, and Elizaveta Filippskikh. 2020. Automatically ranked russian paraphrase corpus for text generation. *arXiv preprint arXiv:2006.09719*.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Chaitra Hegde and Shrikumar Patil. 2020. Unsupervised paraphrase generation using pre-trained language models. *arXiv preprint arXiv:2006.05477*.

Dmitry Kravchenko. 2017. Paraphrase detection using machine translation and textual similarity algorithms. In *Conference on artificial intelligence and natural language*, pages 277–292. Springer.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language. *arXiv preprint arXiv:1905.07213*.

Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020. Pre-training via paraphrasing. *arXiv preprint arXiv:2006.15020*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Phillip Lippe, Pengjie Ren, Hinda Haned, Bart Voorn, and Maarten de Rijke. 2020. Diversifying task-oriented dialogue response generation with prototype guided paraphrasing. *arXiv preprint arXiv:2008.03391*.

Nitin Madnani and Bonnie J Dorr. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*, 36(3):341–387.

Models	Examples
	Скрипач Большого театра погиб после падения в оркестровую яму.
rugpt2	- Скрипач Большого театра скончался после падения в оркестровую яму. (+) - Скрипач Большого театра упал в оркестровую яму и скончался. (+) - Скрипач Большого театра упал в оркестровую яму и умер. (+) - Скрипач Большого театра упал в оркестровую яму и умер от падения. (+-)
rugpt3	- В Большом театре после падения в оркестровую яму погиб скрипач. (+) - В Большом театре подтвердили смерть скрипача от падения в оркестровую яму. (extra info) - При падении в оркестровую яму в Большом театре погиб скрипач. (+) - Упавший в оркестровую яму Скрипач Большого театра умер в больнице. (extra info)
mt5-base	- Более 200 человек погибли при падении в оркестрову яму. (-) - В Большом театре погиб скрыпач после падения в оркестровую яму. (typo) - Скрипач "Большого театра" погиб при падении в оркестровую яму. (+) - Скрипач Большого театра погиб при падении в оркестровую яму. (+)

Figure 2: Generated examples for sentence *Bolshoi Theater violinist dies after falling into orchestra pit*. + is a good generated paraphrase, - is not appropriate paraphrase, *extra info* means the generally good paraphrase but it contains more information than the reference ones.

Marie Meteer and Varda Shaked. 1988. Strategies for effective paraphrasing. In *Coling Budapest 1988 Volume 2: International Conference on Computational Linguistics*.

Tong Niu, Semih Yavuz, Yingbo Zhou, Huan Wang, Nitish Shirish Keskar, and Caiming Xiong. 2020. Unsupervised paraphrase generation via dynamic blocking. *arXiv preprint arXiv:2010.12885*.

Lidia Pivovarova, Ekaterina Pronoza, Elena Yagunova, and Anton Pronoza. 2017. Paraphraser: Russian paraphrase corpus and shared task. In *Conference on Artificial Intelligence and Natural Language*, pages 211–225. Springer.

Ekaterina Pronoza, Elena Yagunova, and Anton Pronoza. 2015. Construction of a russian paraphrase corpus: unsupervised paraphrase extraction. In *Russian Summer School in Information Retrieval*, pages 146–157. Springer.

Lihua Qian, Lin Qiu, Weinan Zhang, Xin Jiang, and Yong Yu. 2019. Exploring diverse expressions for paraphrase generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3164–3173.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Anna Rogers, Alexey Romanov, Anna Rumshisky, Svitlana Volkova, Mikhail Gronas, and Alex Gribov. 2018. Rusentiment: An enriched sentiment analysis dataset for social media in russian. In *Proceedings of the 27th international conference on computational linguistics*, pages 755–763.

Tatiana Shavrina, Alena Fenogenova, Anton Emelyanov, Denis Shevelev, Ekaterina Artemova, Valentin Malykh, Vladislav Mikhailov, Maria Tikhonova, Andrey Chertok, and Andrey Evlampiev. 2020. Russiansuperglue: A russian language understanding evaluation benchmark. *arXiv preprint arXiv:2010.15925*.

AB Siddique, Samet Oymak, and Vagelis Hristidis. 2020. Unsupervised paraphrasing via deep reinforcement learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1800–1809.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2020. Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. *arXiv preprint arXiv:2010.08240*.

Sam Witteveen and Martin Andrews. 2019. Paraphrasing with large language models. *arXiv preprint arXiv:1911.09661*.

Silei Xu, Sina J Semnani, Giovanni Campagna, and Monica S Lam. 2020. Autoqa: From databases to qa semantic parsers with only synthetic training data. *arXiv preprint arXiv:2010.04806*.

Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. In *Proceedings of COLING 2012*, pages 2899–2914.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.