Using RuGPT3-XL Model for RuNormAS competition

Anton Emelyanov[1,2] **Oleh Shliazhko**[1] **Nadezhda Katricheva**[1] **Tatiana Shavrina**[1,3,4]
login-const@mail.ru, olehshliazhko@gmail.com
n.katricheva@gmail.com, rybolos@gmail.com
[1]SberDevices, Sberbank, Moscow, Russia
[2]Moscow Institute of Physics and Technology, Moscow, Russia
[3]National Research University Higher School of Economics, Moscow, Russia
[4]ANO «AI Research Institute», Moscow, Russia

Abstract

The paper presents a fine-tuning methodology of the RuGPT3-XL (Generative Pretrained Transformer-3 for Russian) language model for the normalization of text spans task. The solution is presented in a competition for two tasks: Normalization of Named Entities (Named entities) and Normalization of a wider class of text spans, including the normalization of different parts of speech (Generic spans).
The best solution has achieved 0.9645 accuracy on the Generic spans task and 0.9575 on the Named entities task.
The presented solutions are in the public domain at https://github.com/RussianNLP/RuNormAS-solution

Keywords: text normalization, text generation, evaluation track, ruGPT-3, generative pretrained transformer
DOI: 10.28995/2075-7182-2021-20-204-212

Использование RuGPT3-XL модели для соревнования RuNormAS

Антон Емельянов[1,2] (login-const@mail.ru) Олег Шляжко[1] (olehshliazhko@gmail.com)
Надежда Катричева[1] (n.katricheva@gmail.com) Татьяна Шаврина[1,3,4] (rybolos@gmail.com)

[1]SberDevices, Сбербанк, Москва, Россия
[2]Московский физико-технический институт, Москва, Россия
[3]НИУ «Высшая Школа Экономики», Москва, Россия
[4]АНО «Институт Искусственного Интеллекта», Москва, Россия

Аннотация

В статье представлена методология дообучения языковой модели RuGPT3-XL (Generative Pretrained Transformer-3 для русского языка) для задачи нормализации спанов текста. Решение представлено на конкурсе по двум задачам: Нормализация именованных сущностей (Named entity) и Нормализация более широкого класса фрагментов текста, включая нормализацию различных частей речи (Generic spans).
Лучшее решение достигло точности 0.9645 для задачи нормализации фрагментов текста и 0.9575 для именованных сущностей.
Представляемые решения находятся в открытом доступе по адресу https://github.com/RussianNLP/RuNormAS-solution
Ключевые слова: нормализация текстов, генерация текстов, ruGPT-3, generative pretrained transformer

1 Introduction

The task of normalization is indispensable in Natural Language Processing (NLP) because it allows both to obtain a connection between the wordforms of the same paradigm and to reduce vocabulary size while preserving lexical meaning. Text classification, clusterization, topic modeling, style detection, and many more NLP tasks depend on normalization as a basic stage in the text processing pipeline. Regarding an isolating, fusional or agglutinative morphology type, normalization comes in two basic

wordform procedures: stemming or lemmatization. As a more simplistic approach, stemming only chops word endings from the stem, and thus it often attributes the same stem to cognates or different stems to the same lexeme. Lemmatization, in contrast, aims to bring tokens to lexemes. There are other types of normalization, too, such as expanding contractions and abbreviations, but in this paper, we understand normalization differently due to the entities it is applied to. To normalize a named entity or a phrase means to reduce it to its so-called «initial» form representing the semantic core which stays the same no matter what inflections its constituent parts may adopt to provide the syntactic integrity of a sentence. Most of the time it includes lemmatization, like in the case of the named entity (and noun phrase) "группы компаний ЛУКОЙЛ", which is normalized into "группа компаний ЛУКОЙЛ": the head of the noun phrase, "группа компаний ЛУКОЙЛ", becomes a lexeme after normalization. Proper nouns as parts of named entities can be normalized and plural at the same time, like in the case of "Сердца России", and there are other proper nouns which remain inflected and should not be changed through normalization.

Normalization methods include the usage of lexical databases, where word forms are linked to their lexemes. The result improves if part-of-speech (POS) tags are attributed to the word forms. Another common approach is rule-based, and of course words, as well as named entities and phrases, can be normalized using Neural Networks (NNs) (not only through using NNs for POS-tagging).

This paper is structured as follows: in section 2, we present the already existing research works related to the topic under discussion; section 3 gives a general overview of the competition; section 4 is devoted to our solution of the RuNormAS competition; section 5 provides error analysis, and the paper is concluded in section 6.

2 Previous Work

The English language was traditionally the first to undergo normalization algorithms, in particular, became the object for the first stemmer algorithm [1]. The analytical morphological structure was the best suited for this type of algorithms (for example, [2]), which, together with the growing needs of information retrieval, pushed their development — this happened, in particular, in the works [3], [4]. Nevertheless, it was the fusional and agglutinative languages, with their more productive morphology, that pushed normalization technologies to new levels and made them a subject of competition. Thus, the CONLL competition held in 2016-2018 [5,6, 7] set the task of complete grammatical annotation, from raw text to syntax, which comprised lemmatization for 103 languages, including "surprise languages" in the private test set. For the Russian language, the quality of word inflexion in context achieved 94.4% accuracy.

As for the Russian language separately, normalization technologies are also actively developing for it as a language with a developed morphology. The needs for information retrieval [9] prompted the use of the rich heritage of morphological description [8].

In 2010, for the first time, a shared task was held for automatic Russian part-of-speech tagging, lemmatization, and morphological analysis, including the subtask of annotating rare words [10]. The participants achieved 98.1% accuracy on lemmatization, the test set being not very large. At the MorphoRuEval-2017 shared task [11], a 96.91% accuracy score in lemmatization was achieved on a balanced set of data from various sources (news, social networks, fiction, etc.). And in the GramEval-2020 shared task [12] the track became even more complicated since data from social media, poetry and historical texts of the 17th century were added to the test sample: the best overall lemmatization score being 98% on fiction texts, 98.2% on the news, 95.3% on poetry, 96% on social media, 93% on wiki and 78.3% on historical texts. It became manifest that it is technically possible for the Russian language to pose more complex challenges, especially for notoriously "difficult-to-process" groups of words and lexical categories.

3 Dialog Evaluation 2021 Track

Within the framework of the RuNormAS (Russian Normalization of Annotated Spans) competition [13], the normalization problem is proposed — bringing a part of the text (a named entity, a phrase) to its

normal (initial) form. The main part of the task is to correctly normalize the words from the group that need normalization without changing the other ones (dependent, etc.) while using the given context to the benefit of this task. The latter is especially important since the initial form for many words can be determined only in context — for example, the the word "Иванова", depending on the surrounding context, can have either the normal form "Иванова"or "Иванов".
The competition offers two tracks:

1. Normalizing Named Entities
2. Normalization of a wider class of text spans, including the normalization of different parts of speech.

The data for the first track were collected from the articles of the «ВЗГЛЯД»newspaper, for the second one — from the documents of the Ministry of Economic Development. Both samples were labeled manually.

The quality metric for the task is the percentage of exact matches between the normalization result and the reference.

3.1 Dataset

Both tasks have the same data format. The `text_and_ann` folder contains files with texts (`.txt`) and files with span markup (`.ann`). In the file with the markup, the indices of the beginning and end of the entity in the text are written on each line. If the entity has breaks, then one line is written with the start and end indices for each chunk (and the chunks may be unordered). For example, if an entity has two breaking chunks, then the annotations on the corresponding line will contain `start1 end1 start2 end2` or `start2 end2 start1 end1`. In the folder norm, on each line, there is the result of normalization of the corresponding span. The match is made by the filename up to a dot. Also, for best model additional data was used. We add the "lenta news" dataset to the train data. This is a corpus of Russian News for the year 2019. That corpus was annotated automatically and is a part of Taiga corpus [14].

4 Approach

4.1 Baseline

The competition presents a baseline obtained using normalization tools from the Natasha library[1]. This solution is completely rule-based.

4.2 Neural Language modeling

The idea of finetuning a pretrained Language Model (LM) is at the core of our approach. All the experiments were carried out using RuGPT3XL[2]. The main difference is connected with data preparation for the RuGPT3XL LM finetuning procedure and model inference strategy. We do not separate data for two tasks and train one model at each approach on the whole set of train data.
The main algorithm for making predictions consists of three steps (all of them are described below):

1. Prepare data for LM using one of data preparation approaches;
2. Make predictions with LM using one of the inference strategies;
3. Apply the post-processing pipeline;

Each approach differs from the other one only in a specific template for generation, which is fed to the input of the LM. We tested the following approaches of data preparation (for the first step):

1. Model0 — only left context LM;
2. Model1 — only left context LM with `<start>` special token;
3. Model2 — left and right contexts LM with `<start>` and `<end>` special tokens;
4. Model3 — left and right contexts LM with `<start>` and `<end>` special tokens and additional training data;

For each approach, we apply two inference strategies:

[1] `https://github.com/natasha/natasha`
[2] `https://huggingface.co/sberbank-ai/rugpt3xl`

- **“argmax“ inference strategy** is the decoding strategy of LM. We select the next token by applying ‘argmax‘ operation over probability distribution that is produced by LM on each decoding step.
- **“beam search“ inference strategy** is the standard beam search algorithm with the number of beams equal to 10.

For each approach, we apply the same post-processing pipeline.

4.2.1 Post-processing pipeline

The post-processing pipeline should correct errors that occur while generating with LM (after the second step). We have categorized the errors as follows:

1. extra special tokens — model generates extra special tokens that should be removed;
2. letters case errors — model generates words in different cases;
3. extra or removed punctuation — model generate additional punctuation marks or remove some punctuation;
4. different word count in annotation and prediction;
5. symbol intersection error — this error occurs if the following condition is met:
   $\frac{|set(annotation) \cap set(generated)|}{|set(generated)|} < 0.6$
   here, the annotation and generation are strings; the 0.6 parameter is selected with some greed search on a subsample of the training data.

For steps 4-5 of this pipeline, we get prediction from the baseline model if errors occurred. Other steps of post-processing are also implemented in our repository.

4.2.2 Model0 — only left context LM

For each line in files with span markup (`.ann`), we find a substring in the text that should be normalized. For example, we have a text in the file of the test set with the name “723362“ at Named Entities task:

“Пушков назвал выход Греции из еврозоны «сильнейшим ударом по ЕС» Греция — ключ к будущему ЕС, ее выход из еврозоны станет сильнейшим ударом по ЕС за всю его историю, заявил глава комитета Госдумы по международным делам Алексей Пушков. «Что бы ни говорили в Берлине, выход Греции из еврозоны станет сильнейшим ударом по ЕС за всю его историю. Сегодня Греция — ключ к будущему ЕС», — написал Пушков в Twitter. Накануне пресс-секретарь Еврокомиссии (ЕК), отвечая на вопросы о Греции, заявил, что участие стран в еврозоне согласно законодательству Евросоюза не подлежит отмене. Ранее в издании Der Spiegel сообщалось, что канцлер ФРГ Ангела Меркель и министр финансов страны Вольфганг Шойбле допускают выход Греции из еврозоны при необходимости. Позже заместитель официального представителя кабмина ФРГ Георг Штрайтер заявил, что позиция Германии по вопросу выхода Греции из еврозоны не изменилась. не изменилась“.

For the 12th line, in the annotation file we extract the subtext that needs to be normalized: “Вольфганг Шойбле“. For training LM, we construct a training record with the help of the following template:

```
<s>{left_context}{to_norm}<answer>{norm}</s>
```

Here the `<s>` token denotes beginning of text; `left_context` denotes all text before subtext that should be normalized (to_norm); the `<answer>` token separates input text prefix and answer that LM should learn; norm is the normalized text; and `</s>` token denotes the end of text. For our previous example, we have the following training record:

<s>Пушков назвал выход Греции из еврозоны «сильнейшим ударом по ЕС» Греция — ключ к будущему ЕС, ее выход из еврозоны станет сильнейшим ударом по ЕС за всю его историю, заявил глава комитета Госдумы по международным делам Алексей Пушков. «Что бы ни говорили в Берлине, выход Греции из еврозоны станет сильнейшим

ударом по ЕС за всю его историю. Сегодня Греция — ключ к будущему ЕС», — написал Пушков в Twitter. Накануне пресс-секретарь Еврокомиссии (ЕК), отвечая на вопросы о Греции, заявил, что участие стран в еврозоне согласно законодательству Евросоюза не подлежит отмене. Ранее в издании Der Spiegel сообщалось, что канцлер ФРГ Ангела Меркель и министр финансов страны Вольфганг Шойбле<answer>Вольфганг Шойбле</s>

For the inference phase, we construct the following template that is passed to LM:

<s>{left_context}{to_norm}<answer>.

After making predictions, we correct the output with the post-processing pipeline that is described later.

4.2.3 Model1 — only left context LM with the <start> special token

We use the following template in this data preparation approach:

<s>{left_context}<start>{to_norm}<answer>{norm}</s>

The main difference from the previous template is the token <start> which denotes the beginning of the subtext that should be normalized. For our previous example, we have the following training record:

<s>Пушков назвал выход Греции из еврозоны «сильнейшим ударом по ЕС» Греция — ключ к будущему ЕС, ее выход из еврозоны станет сильнейшим ударом по ЕС за всю его историю, заявил глава комитета Госдумы по международным делам Алексей Пушков. «Что бы ни говорили в Берлине, выход Греции из еврозоны станет сильнейшим ударом по ЕС за всю его историю. Сегодня Греция — ключ к будущему ЕС», — написал Пушков в Twitter. Накануне пресс-секретарь Еврокомиссии (ЕК), отвечая на вопросы о Греции, заявил, что участие стран в еврозоне согласно законодательству Евросоюза не подлежит отмене. Ранее в издании Der Spiegel сообщалось, что канцлер ФРГ Ангела Меркель и министр финансов страны<start>Вольфганг Шойбле<answer>Вольфганг Шойбле</s>

For inference phase, we construct the following template that is passed to LM:

<s>{left_context}<start>{to_norm}<answer>.

After prediction we correct output with post-processing pipeline that described later.

4.2.4 Model2 — left and right contexts LM with the <start> and <end> special tokens

We use the following template in this data preparation approach:

<s>{left_context}<start>{to_norm}<end>{right_context}<answer>{norm}</s>

Here, the <s> token denotes the beginning of the text; left_context denotes the text before the subtext that should be normalized (to_norm); the token <start> denotes the beginning of the subtext that should be normalized; the <end> token denotes the end of the subtext that should be normalized; right_context denotes the text after the subtext that should be normalized (to_norm); the <answer> token separates the input text prefix and the answer that LM should learn; norm is the normalized text; and the </s> token denotes the end of the text. For our previous example, we have the following training record:

<s>написал Пушков в Twitter. Накануне пресс-секретарь Еврокомиссии (ЕК), отвечая на вопросы о Греции, заявил, что участие стран в еврозоне согласно законодательству

Model name	Generic spans	Named entities
Model0 + argmax + not_fixed	0.6953	0.7513
Model0 + argmax	0.7507	0.7891
Baseline	0.7732	0.8881
Model0 + beam search	0.8454	0.8828
Model1 + argmax	0.9059	0.9306
Model1 + beam search	0.9483	0.9455
Model2 + beam search	0.9592	0.9570
Model3 + beam search + not_fixed	0.9636	0.9522
Model3 + beam search	**0.9645**	**0.9575**

Table 1: Evaluation results

Евросоюза не подлежит отмене. Ранее в издании Der Spiegel сообщалось, что канцлер ФРГ Ангела Меркель и министр финансов страны <start> Вольфганг Шойбле <end> допускают выход Греции из еврозоны при необходимости. Позже заместитель официального представителя кабмина ФРГ Георг Штрайтер заявил, что позиция Германии по вопросу выхода Греции из еврозоны не изменилась.<answer>Вольфганг Шойбле</s>

`left_context` and `right_context` are the texts that are limited to 40 words taken before `to_norm` and 40 words after `to_norm`. The parameter of window 40 is selected with some greed search on a subsample of training data.
For the inference phase, we construct the following template that is passed to LM:

```
<s>{left_context}<start>{to_norm}<end>{right_context}<answer>.
```

After prediction, we correct the output with the post-processing pipeline described below.

4.2.5 Model3 — left and right contexts LM with the <start> and <end> special tokens and additional training data

The main difference from the previous approach is using additional training data. We add the "lenta news" corpus with normalization markup and finetune the model on this corpus joint with the training data. After that, we finetune the model only on the training data. The data for LM finetuning was prepared as described in the previous section.

4.2.6 Training details

Each model was trained on 16 GPU with distributed training for around 12 hours. We use the Adam optimizer from [18] with the decoupled weight decay regularization 1e-2 [19]. We use a constant learning rate, 0.000015 on 20000 train iterations with fp16 precision and deepspeed code optimizations[20]. The final perplexity on all models is around 1.0002-1.0005.

5 Error Analysis and Results

5.1 Results

The results of our experiments on test set are presented in Table 1. The best result (*Accuracy Generic spans = 0.9645* and *Accuracy Named entities = 0.9575*) was obtained for "Model3 — left and right contexts LM with <start> and <end> special tokens and additional training data" approach with the beam search inference strategy.

The fourth approach "Model3 — left and right contexts LM with <start> and <end> special tokens and additional training data" with the beam search inference strategy obtains the best accuracy for the RuGPT3XL model in this competition. Also, we can see the difference provided by the post-processing pipeline on "Model0" and "Model3". For the last model, the impact is minor because the LM model has very strong results and sees more data.

5.2 Error analysis

5.2.1 Evaluation errors

Here, we describe errors that are connected with the incorrect data in the evaluation set and markup. We have categorized errors into the following classes:

1. word count errors — these errors denote different counts of words in gold prediction and annotation. For example: "Костромская область"and "областях", here our best model predicted "области".
2. titled errors — these errors denote difference between word cases in gold prediction and annotation. For example: "Генпрокуратура Украины"and "генпрокуратура Украины", here our best model predicted "генпрокуратура Украины".
3. symbol errors — these errors denote the difference between some symbols in gold prediction and annotation. For example: "город Антрацит"and "город Антрацит", here our best model predicted "город Антрацит".
4. punctuation errors — these errors denote the difference between the punctuation in gold prediction and annotation. For example: "ООО «Первая топливная компания»"and "ООО Первая топливная компания», here our best model predicted "ООО Первая топливная компания»".
5. word start errors — these errors denote the difference between the starting symbols in gold prediction and annotation. For example: "расти"and "будет расти", here our best model predicted "будет расти". Also these errors denote encoding mismatch or truncated markup.

If these errors are not taken into account, then the model obtained **0.9767** accuracy on the Generic spans task and **0.9810** accuracy on the Named entities task.

5.2.2 Model errors

Here we describe model errors. We divide the errors into categories:

1. word count errors. An example of prediction and gold prediction: "дорога Артемовск-Луганское-Дебальцево"and "дорога Артемовск-Луганское-Лозовое-Дебальцево".
2. word position errors. An example of prediction and gold prediction: "Киевская городская государственная администрация" and "Киевская государственная городская администрация".
3. word ending errors. An example of prediction and gold prediction: "Верховная рада"and "Верховая рада".
4. word case errors. An example of prediction and gold prediction: "«взрослый» Арктический Совет"and "«взрослый» Арктический совет".
5. consistency errors. An example of prediction and gold prediction: "Южно-Русский газоконденсатный месторождение"and "Южно-Русское газоконденсатное месторождение".
6. errors with foreign words. An example of prediction and gold prediction: "Укрнафта"and "Укртанснафта".
7. errors with POS tags mismatches. An example of prediction and gold prediction: "Новороссийский"and "Новороссийск".

Some of the described errors can be avoided by finetuning the model with more extra data.

6 Conclusion and Future Work

We present the results of our participation in the DE2021: RuNormAS (Russian Normalization of Annotated Spans) task. The implemented methods in both subtracks are based on RuGPT3XL LM. As future work, we plan to finetune RuGPT3XL LM on more extra data.

The best model was presented in the paper is available open-source. We hope that our developments will be useful to the community since all the presented prototypes are easily portable to new domains and tasks.

References

[1] Lovins, Julie Beth (1968). "Development of a Stemming Algorithm" (PDF). Mechanical Translation and Computational Linguistics. 11: 22–31.
[2] Dawson, J. L. (1974); Suffix Removal for Word Conflation, Bulletin of the Association for Literary and Linguistic Computing, 2(3): 33–46.
[3] Frakes, W. B. (1984); Term Conflation for Information Retrieval, Cambridge University Press.
[4] Frakes, W. B. (1992); Stemming algorithms, Information retrieval: data structures and algorithms, Upper Saddle River, NJ: Prentice-Hall, Inc.
[5] Cotterell R. et al. The SIGMORPHON 2016 shared task—morphological reinflection //Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology. – 2016. – . 10-22.
[6] Cotterell R. et al. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages //arXiv preprint arXiv:1706.09031. – 2017.
[7] Cotterell R., Kirov Ch., Sylak-Glassman J., et al. (2018) The CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection. In Proceedings of CoNLL–SIGMORPHON 2018.
[8] Zaliznyak, A. A. (1977) Grammatical Dictionary of Russian [Grammaticheskij slovar' russkogo yazyka]. Moscow.
[9] Segalovich I. (2003) A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine, In Proceedings of MLMTA-2003, Las Vegas, Nevada, USA.
[10] Lyashevskaya, Olga, Irina Astaf'eva, Anastasia Bonch-Osmolovskaya, Anastasia Garejshina, Julia Grishina, Vadim D'jachkov, Maxim Ionov, Anna Koroleva, Maxim Kudrinsky, Anna Lityagina, Elena Luchina, Eugenia Sidorova, SvetlanaToldova, Svetlana Savchuk, and Sergej Koval' (2010) NLP evaluation: Russianmorphological parsers [Ocenka metodov avtomaticheskogo analiza teksta: morfologicheskije parsery russkogo jazyka]. In: Computational linguistics and in-tellectual technologies. Proceedings of International Workshop Dialogue'2010. Vol. 9 (16), 2010. Pp. 318–326
[11] Sorokin, A., Shavrina, T., Lyashevskaya, O., Bocharov, B., Alexeeva, S., Droganova, K., ... Granovsky, D. (2017). MorphoRuEval-2017: an evaluation track for the automatic morphological analysis methods for Russian.
[12] Lyashevskaya O., Shavrina T., Trofimov I., Vlasova N. A. GRAMEVAL 2020 Shared Task: Russian Full Morphology and Universal Dependencies Parsing, in: Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной международной конференции «Диалог» (Москва, 17 июня — 20 июня 2020 г.) / Под общ. ред.: В. Селегей. Вып. 19(26). М. : Изд-во РГГУ, 2020. P. 553-569.
[13] Zolotukhin, Denis and Smurov, Ivan (2021). RuNormAS-2021: a Shared Task on Russian Normalization of Annotated Spans. Computational Linguistics and Intellectual Technologies: Papers from the Annual Conference "Dialogue".
[14] Shavrina, Tatiana and Shapovalova, Olga (2017). TO THE METHODOLOGY OF CORPUS CONSTRUCTION FOR MACHINE LEARNING:"TAIGA". SYNTAX TREE CORPUS AND PARSER. Corpus linguitics-2017. Pp 78.
[15] Ivanin, V. A., et al. "Rurebus-2020 shared task: Russian relation extraction for business." (2020).
[16] Starostin, A. S., et al. "FactRuEval 2016: evaluation of named entity recognition and fact extraction systems for Russian." (2016).
[17] Malykh, Kalaidin. "HEADLINE GENERATION SHARED TASK ON DIALOGUE'2019." Компьютерная лингвистика и интеллектуальные технологии. 2019.
[18] Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[19] Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983, 2016.
[20] Minjia Zhang, Yuxiong He. (2020) Accelerating Training of Transformer-Based Language Models with Progressive Layer Dropping. arXiv:2010.13369.