RuSimScore: unsupervised scoring function for Russian sentence simplification quality

Mikhail Orzhenovskii
Saint Petersburg, Russia
orzhan057@gmail.com

Abstract

We propose an unsupervised complex scoring function (RuSimScore) to measure simplification quality of Russian sentences, and a model for text simplification based on this function. The function allows to score simplicity and original meaning preservation. First, filtered a noisy parallel corpus (machine translated WikiLarge) and extracted good simplification examples. After that, a pretrained language model was fine-tuned on these examples. We generate multiple outputs from the language model and select the best one according to the scoring function. The weights in the scoring function can be adjusted to balance between better content preservation and getting simpler sentences (controllable simplification).

Keywords: text simplification, pretrained language models, Russian
DOI: 10.28995/2075-7182-2021-20-524-532

RuSimScore: функция для оценки качества упрощения текста на русском языке

Орженовский М.В.
Санкт-Петербург, Россия
orzhan057@gmail.com

Аннотация

Мы предлагаем составную оценочную функцию (RuSimScore) для измерения качества упрощения текстов на русском языке, а также модель, построенную с помощью этой функции. Она позволяет оценить простоту результата и степень сохранения смысла исходного текста. Сначала из зашумленного корпуса (переведенный WikiLarge) отфильтровали примеры упрощения с достаточным качеством. Затем предобученная языковая модель была дообучена на этих примерах. С помощью этой языковой модели мы генерировали множество выходных предложений и выбирали лучшее на основе оценочной функции. Веса оценочной функции можно изменять (контролируемое упрощение), чтобы выбирать между лучшим сохранением смысла или более простыми выходными предложениями.

Ключевые слова: упрощение, предобученные языковые модели, русский язык

1 Introduction

RuSimpleSentEval (RSSE) competition[19] introduced the first text simplification dataset for Russian, which was collected on a crowd-sourcing platform. Another dataset suitable for Russian text simplification is WikiLarge, built from machine translated English Wikipedia and Simple English Wikipedia. WikiLarge has two issues: sometimes inaccurate alignment and errors introduced during machine translation. It has be be filtered in order to obtain high quality sentence pairs for training.

The idea of the proposed scoring function is to combine different aspects of simplification quality in a single-number metric which depends only on source (complex) and target (simplified) sentences, and does not require human labeling (like SARI). It is built from six different simple functions, which will be described later.[1]

[1]The code for training and running the model, as well as best model's weights will be published as open source at https://github.com/orzhan/rusimscore

We did not use the scoring function to directly calculate loss during the model training. Instead, we filtered the machine translated WikiLarge dataset (getting better result than with non-filtered one), extracting 15% of examples. Then we fine-tuned pretrained language model ruGPT-3 on the resulting dataset (sequence to sequence task can be converted to language modeling task by inserting special tokens into prompt). During inference we used the fine-tuned language model to generate multiple answers for each input sentence using nucleus sampling, and then selected the best answer with the scoring function.

2 Related work

Text simplification is often done with a sequence to sequence model trained on parallel corpus. Zhang and Lapata[21] use reinforcement learning of with a task-specific reward function. Martin et al.[3] trained a sequence to sequence model with controllable parameters. They also filtered sentences suitable for simplification from a very large corpus, based on cosine similarity of sentence level embeddings[16] .

Another approach is simplifying text in several consecutive steps. The task can be formalized as sequence labeling [15], generating a sequence of changes using a programmer-interpreter approach [7], or iteratively applying changes to a text [10].

Kuvshinova[12] solved sentence compression task for Russian with deletion based approach. This task is related to simplification.

Readability evaluation is language specific. For Russian, Ivanov et al.[11] identified text features that indicate complex texts. Laposhina et al.[2] explored readability formulas.

Language models can solve a wide range of language processing tasks including abstractive summarization [14] and paraphrase generation [20], these tasks are related to simplification.

Our work uses fine-tuned ruGPT3 language model [13] for text generation.

3 Model description

3.1 Scoring function

The idea of scoring function RuSimScore(c,s) is to estimate if a simple sentence s is a good simplification of the complex sentence c. Good simplification means that the target sentence is simple, and the meaning of the source sentence is preserved. Scoring function is calculated as multiplication of four simplicity scoring functions: lexical complexity score, dependency tree depth score, length score, reading ease score, and two content preservation scoring functions: cosine similarity score, named entity preservation score.

$$RuSimScore(c,s) = LS^{\alpha}(c,s)DD^{\beta}(c,s)LeS^{\gamma}(c,s)RS^{\delta}(c,s)SimS^{\epsilon}(c,s)NS^{\zeta}(c,s)$$

$$RuSimScore \in [0,1]$$

Where $\alpha, \beta, \gamma, \delta, \epsilon, \zeta$ are weights that allow to control importance of different aspects of similarity. In this work we selected the optimal weights which maximized SARI on the development dataset.

Cosine similarity score (SimS) is calculated as cosine similarity between sentence level embeddings of source and target sentences. We have chosen LASER embeddings[1].

Named entity preservation score (NS) aims to help with the weakness of the embeddings: if the language model modifies named entities in the text, the change in the embeddings may be very small, however the meaning of the text can become very different. To calculate named entity score, we extract all named entities from both source and target sentences using Natasha library and calculate how many entities from target are also present in source. While matching entities we only require that one of the entity's words is matched (so that entities themselves can also be simplified like: Оскар Александрович Энгберг $\rightarrow$ Энгберг). If count of entities is greater than 3, then 3 matching entities are considered enough and NS is set to 1.0.

$$NS = \frac{min(3, |NER(c) \cap NER(s)|)}{min(3, |NER(c) \cup NER(s)|)}$$

Lexical complexity score (LS) can judge if the words used in the target sentence are more common in language (which corresponds to higher usage frequency in corpus). Score is calculated as:

$$LS = 1 + \alpha_{LS} \frac{\sum_{i=1}^{N} log(f_i)}{N} + \beta_{LS} min(log(f_i))$$

where f_i - frequency of i-th word. Unknown words, named entities, pronouns and numbers are excluded from the calculation. Lexical complexity score consists of average log frequency with weight α_{LS} and most rare word's log frequency with weight βLS. Word frequency data was taken from [5][2].

Dependency tree depth score (DS) aims to measure syntactical complexity of the target sentence. The syntax tree of the sentence is created using Natasha library[3]. Dependency tree score DD = 1.0 for depth of 1 or 2, DD = 0.9 for depth 3, DD = 0.7 for depth 4 and DD = 0.5 for larger depths.

Length score (LeS) helps to choose target sentences than are shorter than the source one (in terms of word count), but not too short:

$LeS(c, s) = 0.5$ if $WC(s) > WC(c)$

$LeS(c, s) = 1 - \frac{WC(s)}{2WC(c)}$ if $WC(s) > 6$ and $WC(s) \leq WC(c)$

$LeS(c, s) = \frac{WC(s)}{6}$ if $WC(s) \leq 6$ and $WC(s) \leq WC(c)$

where $WC(x)$ is word count in sentence x.

Reading ease score (RS) is an implementation of Flesch reading ease with coefficients for Russian language [17], mapped into [0.5,1] range:

$$RS = 0.75 + 0.25 \frac{max(-100, min(100, 206.835 - 1.52WPS - 65.14\frac{SC}{WC}))}{100}$$

where WPS is number of words per sentence, SC is syllable count and WC is word count.

See Table 1. for examples of scoring function values.

3.2 Generative model

We used ruGPT-3, a GPT-3 implementation by SberBank AI[4]. This language model has received high score on Russian SuperGLUE benchmark[18] and is capable of solving various tasks.

During fine-tuning, the training set is inputted into the LM in the following format:

<s>Original sentence <Simplify:> Target sentence </s>

During inference, for source sentence we provide the following prompt:

<s>Original sentence <Simplify:>

and expect LM to output the simplified sentence and </s> token.

We use top-p sampling[4] to increase fluency and variance of the generated sequences, and generate 10-100 sentences for each original sentence. After that we calculate the scoring function for each of the generated sentences and select the best ones.

Language models are initially trained on next token prediction objective. As a result, sometimes text continuation is generated instead of simplification. To counter this, we had to add checking if the generated sentence starts with a pronoun or a determiner.

We use HuggingFace Transformers implementation to fine-tune the model and generate text[9].

4 Datasets

Machine translated WikiLarge dataset was provided by the competition organizers. We selected 37,884 samples from 246,978 samples of WikiLarge which were filtered by following condition: ES $\geq$ 0.65, SimS $\geq$ 0.75, RS $\geq$ 0.6, LeS $\geq$ 0.55, LS $\geq$ 0.65, DS $\geq$ 0.5. These conditions were based on the statistics of the scoring functions on the RSSE dev dataset (so quality of selected samples was on the same level as quality of RSSE dev samples).

For examples of the selected and removed samples see Table 3.

[2] https://github.com/hermitdave/FrequencyWords

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

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

Таблица 1: Examples of scoring function values

Original sentence: Положение стало угрожающим для царевича, когда Филипп женился в седьмой раз — на знатной македонянке Клеопатре.

Text	RuSimScore	DS	LS	LeS	RS	ES	SimS
Положение стало угрожающим для царевича, когда Филипп женился в седьмой раз — на знатной македонянке Клеопатре.	0.21	0.90	0.69	0.50	0.70	1.00	1.00
Наследник престола не был в восторге от этого брака. Он был счастлив жениться на красивой македонянке, но жениться на египтянке.	0.08	0.90	0.82	0.50	0.85	0.50	0.60
Вскоре после этого царевич Филипп женился на Клеопатре.	0.16	1.00	0.79	0.67	0.80	0.60	0.71
Филипп женился в седьмой раз, на македонянке Клеопатре.	0.32	0.90	0.81	0.67	0.84	1.00	0.86

Table 2: Datasets

Dataset	Reference samples	Simplification samples
RSSE dev	1000	3574
WikiLarge filtered	37884	37884
RSSE public test	1000	3521
RSSE hidden test	1126	N/A

Таблица 3: Selected and dropped samples from WikiLarge

Selected samples:
В Голландии они назывались Stadspijpers, в Германии Stadtpfeifer и в Италии Pifferi. → Их называли Stadtpfeifer в Германии и Pifferi в Италии.
Иногда могут появляться оттенки красного и оранжевого, заменяя или смешиваясь с желтым в зависимости от подвида. → Иногда могут появляться оттенки красного и оранжевого.
Dropped samples:
Женева - второй по численности населения город Швейцарии (после Цюриха) и самый густонаселенный город Романди (франкоговорящая часть Швейцарии). → Он окружен двумя горными цепями - Альпами и Юрой.
Оливковое масло также используется в мыловарении и в качестве лампового масла. → Оливковое масло - это растительное масло.

The final model was trained on both RSSE dev and WikiLarge filtered dataset, its hyperparameters were chosen based on results on RSSE public test dataset, and the final score was obtained on RSSE hidden test dataset.

5 Results and analysis

In the competition, submissions were scored based on SARI[6]. This metric includes F1 score of add, delete and keep operations on n-gram level.

The evaluation results of the proposed model and the benchmarks are shown in Table 4. Iterative deletion is our implementation that is not using a language model; instead it iteratively removes the parts of syntax tree if it increases RuSimScore (partial implementation of [10]). The result of official benchmark is taken from public test.

When more target sentence candidates are generated with the language model, the best candidates are better in terms on SARI. However, generating too many candidate sentences causes drop in score. The scoring function is selecting too short simplifications in this situation. See Table 5.

Different language model sizes are compared in Table 6. Medium model is slightly better then the large one, and both perform better than the small model.

Ablation study of the scoring function is displayed in Table 7. All of the six functions that are included in RuSimScore appear to be useful.

Examples of controllable simplification are shown in Table 8. We can see the effect of modifying the weights of the scoring function. For example, increased SimS weight leads to more accurate but more complex answer, and increased RS weight leads to less accurate but very simple result.

Table 4: Results

Model	Hidden test SARI
ruGPT3 on filtered WikiLarge + RuSimScore	**39.28**
ruGPT3 on filtered WikiLarge	38.68
Official benchmark (mBART)	30.15
Iterative deletion with RuSimScore	32.40
First half of source text	30.33
Source text unchanged	11.04

Table 5: Generated sentence count

Count	Hidden test SARI
100	39.28
30	**39.39**
10	39.16
1	38.68

Table 6: LM size dependency

Model	Hidden test SARI
ruGPT3-small	38.89
ruGPT3-medium	**39.34**
ruGPT3-large	39.28

Table 7: Ablation study of scoring function

Model	Hidden test SARI
Original RuSimScore	**39.28**
RuSimScore - SimS	37.22
RuSimScore - NS	38.94
RuSimScore - LS	39.03
RuSimScore - LeS	38.91
RuSimScore - DS	39.27
RuSimScore - RS	39.11

Таблица 8: Controllable simplification examples

Original sentence	Архимандрит Дионисий торопил ополчение поспешить к Москве и направил князю Трубецкому просьбу объединиться со Вторым ополчением.
Best model (balanced)	Архимандрит Дионисий сказал князю Трубецкому торопиться к Москве.
More accurate (increased SimS and NS weight)	Архимандрит Дионисий призвал ополчение поспешить к Москве и попросил князя Трубецкого объединиться с ними.
Simpler (increased RS, LS, LeS, DS weight)	Архимандрит Дионисий был в Москве и просил войска помочь ему

6 Error analysis

Neural sequence to sequence models sometimes generate factually inconsistent abstractive summaries[8]. Sentence simplification task is related to abstractive summarization, so we can expect similar issues in our model.

We performed a human evaluation of 100 randomly selected simplifications produced by the model. We asked the assessors to evaluate: 1) are the generated simplifications well formed Russian sentences and 2) are they factually consistent with the originals. We evaluated two models: one with default scoring function weights (according to highest SARI) and one with increased SimS and NS weights (to improve accuracy). The results of the evaluation (Table 9) show that 20-30% of the sentences have issues with factual consistency. Examples of the detected issues are shown in Table 10.

Table 9: Evaluation of 100 generated simplified sentences

	Original weights	Better accuracy
Well formed sentences	90	93
Not well formed sentences	10	7
Factually consistent	71	81
Issues with factual consistency, including:	29	19
Omitting significant details	15	7
Wrong entity reference	8	6
Incorrect replacement word used	6	6

Таблица 10: Examples of factual inconsistency

Original sentence	Влияние организованной преступности, наркосиндикатов, мафиозных структур, в особенности — в южных регионах — Кампании, Калабрии и на Сицилии.
Omitting significant details	На юге Кампания, Калабрия и Сицилия.
Original sentence	Лейси активно начал поединок, прессингуя и зажимая Джонса у канатов ринга, но Рой умело уходил от ударов и большинство блокировал.
Wrong entity reference	Рой атаковал Джонса и пытался его прессинговать, но Рой успешно отбивался.
Original sentence	Летописцы указывали, что кони уже не могли не ступать по трупам, так как не было чистого места.
Incorrect replacement word used	Летописцы писали, что кони уже не могли не пачкать трупы.

We also performed human evaluation of the scoring function. For 20 examples we provided 5 different simplifications and asked humans to order them from the most preferred one to the least preferred one. Spearman correlation between ranks of simplifications assigned by human and by the scoring function is 0.53 which indicates moderate correlation. Sample size was small, so this metric needs further evaluation on a larger scale.

7 Conclusion

The proposed model was able to produce higher results than the benchmarks, however the simplifications produced by the model may be inaccurate and therefore require manual verification. Selecting the best generated sentence with the scoring function improves the result, and scoring function is correlated with human perception. A simple iterative deletion approach, guided by the scoring function, was able to outperform the official benchmark. For future research, the scoring function can be used in different setup, for example fine-tuning with reinforcement learning.

References

[1] Artetxe Mikel, Schwenk Holger. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. — 2019. — 1812.10464.

[2] Automated Text Readability Assessment for Russian Second Language Learners / A. Laposhina, T. Veselovskaya, M. Lebedeva, O. Kupreshchenk // Proceedings of the international conference Dialogue 2018. — 2018.

[3] Martin Louis, Sagot Benoît, Éric de la Clergerie, Bordes Antoine. Controllable Sentence Simplification. — 2020. — 1910.02677.

[4] Holtzman Ari, Buys Jan, Du Li et al. The Curious Case of Neural Text Degeneration. — 2020. — 1904.09751.

[5] Dave Hermit. Github: Repository for Frequency Word List Generator and processed files. c2016. — 2016.

[6] EASSE: Easier Automatic Sentence Simplification Evaluation / Fernando Alva-Manchego, Louis Martin, Carolina Scarton, Lucia Specia // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations. — Hong Kong, China : Association for Computational Linguistics, 2019. — Nov. — P. 49–54. — Access mode: https://www.aclweb.org/anthology/D19-3009.

[7] EditNTS: An Neural Programmer-Interpreter Model for Sentence Simplification through Explicit Editing / Yue Dong, Zichao Li, Mehdi Rezagholizadeh, Jackie Chi Kit Cheung // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. — Florence, Italy : Association for Computational Linguistics, 2019. — Jul. — P. 3393–3402. — Access mode: https://www.aclweb.org/anthology/P19-1331.

[8] Cao Ziqiang, Wei Furu, Li Wenjie, Li Sujian. Faithful to the Original: Fact Aware Neural Abstractive Summarization. — 2017. — 1711.04434.

[9] Wolf Thomas, Debut Lysandre, Sanh Victor et al. HuggingFace's Transformers: State-of-the-art Natural Language Processing. — 2020. — 1910.03771.

[10] Iterative Edit-Based Unsupervised Sentence Simplification / Dhruv Kumar, Lili Mou, Lukasz Golab, Olga Vechtomova // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. — Online : Association for Computational Linguistics, 2020. — Jul. — P. 7918–7928. — Access mode: https://www.aclweb.org/anthology/2020.acl-main.707.

[11] Ivanov V., Solnyshkina M., Solovyev V. Efficiency of Text Readability Features in Russian Academic Texts // Proceedings of the international conference Dialogue 2018. — 2018.

[12] Kuvshinova T. Sentence Compression for Russian: Dataset and Baselines // Proceedings of the international conference Dialogue 2020. — 2020.

[13] Brown Tom B., Mann Benjamin, Ryder Nick et al. Language Models are Few-Shot Learners. — 2020. — 2005.14165.

[14] Language Models are Unsupervised Multitask Learners / Alec Radford, Jeff Wu, Rewon Child et al. — 2019.

[15] Learning How to Simplify From Explicit Labeling of Complex-Simplified Text Pairs / Fernando Alva-Manchego, Joachim Bingel, Gustavo Paetzold et al. // Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers). — Taipei, Taiwan : Asian Federation of Natural Language Processing, 2017. — Nov. — P. 295–305. — Access mode: https://www.aclweb.org/anthology/I17-1030.

[16] Martin Louis, Fan Angela, Éric de la Clergerie et al. Multilingual Unsupervised Sentence Simplification. — 2020. — 2005.00352.

[17] Oborneva Irina. Mathematical model of the estimation of educational texts[Matematicheskaja model' ocenki uchebnyh tekstov.] // Proceedings of the 15th international conference on Information Technologies in Education (ITO-2005) [Materialy XV Mezhdunarodnoj konferencii-vystavki Informacionnye tehnologii v obrazovanii (ITO-2005)]. — Moscow, 2005.

[18] Shavrina Tatiana, Fenogenova Alena, Emelyanov Anton et al. RussianSuperGLUE: A Russian Language Understanding Evaluation Benchmark. — 2020. — 2010.15925.

[19] Sakhovskiy Andrey; Izhevskaya Alexandra; Pestova Alena; Tutubalina Elena; Malykh Valentin; Smurov Ivan; Artemova Ekaterina. RuSimpleSentEval-2021 Shared Task: Evaluating Sentence Simplification for Russian // Computational Linguistics and Intellectual Technologies: Papers from the Annual Conference "Dialogue". — Vol. XX. — 2021. — P. xx–xx.

[20] Witteveen Sam, Andrews Martin. Paraphrasing with Large Language Models // Proceedings of the 3rd Workshop on Neural Generation and Translation. — Hong Kong : Association for Computational Linguistics, 2019. — Nov. — P. 215–220. — Access mode: `https://www.aclweb.org/anthology/D19-5623`.

[21] Zhang Xingxing, Lapata Mirella. Sentence Simplification with Deep Reinforcement Learning // Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. — Copenhagen, Denmark : Association for Computational Linguistics, 2017. — Sep. — P. 584–594. — Access mode: `https://www.aclweb.org/anthology/D17-1062`.