Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference “Dialogue 2023”

Moscow, June 14–17, 2023

Pseudo-Labelling for Autoregressive Structured Prediction in Coreference Resolution

Vladislav Bolshakov
NTR Labs, Moscow, Russia
BMSTU, Moscow, Russia
vbolshakov@ntr.ai

Nikolay Mikhaylovskiy
NTR Labs, Moscow, Russia
Higher IT School, Tomsk State University, Tomsk, Russia
nickm@ntr.ai

Abstract

Coreference resolution is an important task in natural language processing, since it can be applied to such vital tasks as information retrieval, text summarization, question answering, sentiment analysis and machine translation. In this paper, we present a study on the effectiveness of several approaches to coreference resolution, focusing on the RuCoCo dataset as well as results of participation in the Dialogue Evaluation 2023. We explore ways to increase the dataset size by using pseudo-labelling and data translated from another language. Using such technics we managed to triple the size of dataset, make it more diverse and improve performance of autoregressive structured prediction (ASP) on coreference resolution task. This approach allowed us to achieve the best results on RuCoCo private test with increase of F1-score by 1.8, Precision by 0.5 and Recall by 3.0 points compared to the second-best leaderboard score. Our results demonstrate the potential of the ASP model and the importance of utilizing diverse training data for coreference resolution.

Keywords: Pseudo-Labelling, Autoregressive Structured Prediction, Coreference Resolution
DOI: 10.28995/2075-7182-2023-22-26-33

Псевдоразметка для разрешения кореферентности при использовании авторегрессионного структурированного предсказания

Владислав Большаков
ООО «НТР», Москва, Россия
МГТУ им. Баумана, Москва, Россия
vbolshakov@ntr.ai

Николай Михайловский
ООО «НТР», Москва, Россия
Высшая ИТ-Школа Томского Государственного Университета, Томск, Россия
nickm@ntr.ai

Аннотация

Разрешение кореферентности является важной задачей в области обработки естественного языка, поскольку она используется как элемент решения таких задач, как поиск информации, суммаризация текста, ответы на вопросы по тексту, анализ тональности текста и машинный перевод. В данной статье исследована эффективность различных подходов к разрешению кореферентности на русском языке, с фокусом на набор данных RuCoCo. Также представлены результаты участия в Dialogue Evaluation 2023. Исследованы способы увеличения размера набора данных с помощью псевдоразметки и перевода данных с другого языка. Используя такой подход, удалось утроить размер набора данных, сделать его более разнообразным и улучшить результаты авторегрессионного структурированного предсказания в задаче разрешения кореферентности. Такой подход позволил добиться наилучших результатов на частном тестовом наборе RuCoCo с повышением F1-меры, точности и полноты на 1.8, 0.5 и 3.0 процентных пункта соответственно по сравнению со вторым лучшим результатом. Наши результаты демонстрируют потенциал модели ASP и важность использования разнообразных обучающих данных для разрешения кореферентности на русском языке.

Ключевые слова: псевдоразметка, авторегрессионное структурированное предсказание, разрешение кореферентности

1 Introduction

1.1 Coreference Resolution

Coreference resolution is a natural language processing (NLP) task that involves identifying all the expressions in a text that refer to the same entity or concept, and then linking them together. It is typically modeled by identifying entity mentions (contiguous spans of text), and predicting an antecedent mention for each span that refers to a previously-mentioned entity, or a null-span otherwise. The goal is to determine which pronouns, nouns, and other expressions in a sentence or document refer to the same entity, and to group them into clusters accordingly. Coreference resolution is a challenging task because it requires good understanding of the context and the ability to recognize complex relationships between words and phrases. However, this task is crucial in many applications of NLP, such as information retrieval [1], text summarization [2], question answering [3], sentiment analysis [4] and machine translation [5]. In addition, coreference resolution can be used to improve the readability of a text, by replacing repeated mentions of the same entity with a pronoun or other reference.

1.2 Related Work

This section contains a brief overview of previous most recent coreference resolution models. Lee et al. [6] proposed an end-to-end model for coreference resolution that predicts an antecedent probability distribution over candidate spans. The model incorporates mention scores, coarse and fine coreference scores, and vector representations of the spans to learn a probability distribution over all possible antecedent spans for each span in the text. To improve computational efficiency while being competitive with other models Kirstain et al. [7] introduced a lightweight end-to-end coreference model that removes the dependency on span representations. Instead, they utilize the endpoints of a span (rather than all span tokens) to compute the mention and antecedent scores. But this approach still presents a computational challenge of $O(n^4)$ complexity over document length so the authors need to prune the resulting mentions. Dobrovolskii [8] considers coreference links between words instead of spans which reduces the complexity of the coreference model to $O(n^2)$ and allows it to consider all potential mentions without pruning any of them out. Instead of using mention or coreference scorer within search algorithms over possible spans of text, Bohnet et al. [9] proposed fundamentally different approach that uses a text-to-text (seq2seq) paradigm to predict mentions and links jointly. The T5-based model takes a single sentence as input, and outputs an action corresponding to a set of coreference links involving that sentence as its output. Liu T. et al. [10] proposed another seq2seq T5-based model for Autoregressive Structured Prediction, which is described in more detail in the next section.

1.3 Autoregressive Structured Prediction

Autoregressive Structured Prediction (ASP) represents structures as sequences of actions, which build pieces of the target structure step by step. For instance, in the task of coreference resolution, the actions build spans (contiguous sequences of tokens) as well as the relations between the spans.

The goal of ASP is to predict an action sequence $y = y_1, \ldots, y_N$, where each action y_n is chosen from an action space $\mathcal{Y}_n$ represented as $\mathcal{Y}_n \stackrel{\text{def}}{=} \mathcal{A} \times \mathcal{B}_n \times \mathcal{Z}_n$, where $\mathcal{A}$ is a set of structure-building actions, $\mathcal{B}_n$ is the set of bracket-pairing actions, and $\mathcal{Z}_n$ is a set of span-labeling actions.

The set of structure-building actions $\mathcal{A} = \{$`]`, `[*`, `copy`$\}$ allows to encode the span structure of a text, e.g., `[*Delaware]` encodes that Delaware is a span of interest. Specifically, the action `]` refers to a right bracket that marks the right-most part of a span. The action `[*` refers to a left bracket that marks the left-most part of a span. The superscript * on `[*` indicates that it is a placeholder for 0 or more consecutive left brackets. Finally, `copy` refers to copying a word from the input document. To see how these actions come together to form a span, consider the string `[*Delaware]`, which is generated from a sequence of structure-building actions `[*` , copy , and `]` and the input string `Delaware`.

The set of bracket-pairing actions consists of all previously constructed left brackets, i.e.:

$$\mathcal{B}_n = \{m | m < n \wedge a_m = \text{[*}\}$$

Thus, in general, $|\mathcal{B}_n|$ is $O(n)$. However, it is often the case that domain-specific knowledge can be used to prune $\mathcal{B}_n$. For instance, coreference mentions and named entities rarely cross sentence boundaries, which yields a linguistically motivated pruning strategy [11].

For the task of coreference resolution, the set of span-labelling actions is

$$\mathcal{Z}_{\mathrm{n}} = \{m | m < n \wedge a_m = \blacksquare\} \cup \{\epsilon\}$$

where ϵ by the convention set in [12] is the antecedent of the first mention in each coreference chain and $\{m | m < n \wedge a_m = \blacksquare\}$ is the set of all the previous spans, which allows the model to capture intra-span relationships.

The coreference structure built on top of a document D is first converted into an action sequence and then is modelled as a conditional language model

$$p_\theta(y|D) = \prod_{n=1}^{N} p_\theta(y_n | y_{<n}, D)$$

The model is built on the base of a pre-trained language model such as T5.

2 Preliminary Experiments and Baselines

During the competition we tested several approaches:

- SpaCy implementation[1] of coarse-to-fine model [8] with different backbone models;
- Different transformers pretrained with Longformer [13] architecture;
- Original implementation[2] of start-2-end model [7] with different backbone models;
- Original implementation[3] of ASP [10] with different backbone models.

2.1 SpaCy

We trained a word-level spacy-coref model on RuCoCo dataset with different transformer encoders. It is trained in two stages: coreference clustering model that use coarse and fine scores to form clusters of entities, than span resolution model that recover original span after word-level coreference resolution. The best backbone transformer model was cointegrated/LaBSE-en-ru[4]. Although this model is a great sentence encoder, it did a good job on the word-level task too. This approach managed to beat baseline of the competition (a 2.4 point higher F1-score).

2.2 Longformers

Since the documents in RuCoCo dataset are relatively long we considered Longformer models that are able to grasp a larger area of text and its context. We pretrained two Longformer models that were based on cointegrated/LaBSE-en-ru and sberbank-ai/ruRoberta-large[5]. Pretraining was done according to [13] using long documents from Russian part of Wikipedia. Using these models together with spacy-coref and increased input sequence length barely gave us a performance gain, while making models even more memory intensive.

3 Improving Autoregressive Structured Prediction Performance

To beat the results of our previous best model we used ASP without changes in the implementation. With that said, we can divide further improvements of the model in two parts:

1. Choosing the backbone model along with hyperparameters tuning;
2. Working on the dataset improvement.

While experimenting with ASP we used different ruT5 models, different training sequence lengths and hidden sizes of the structure-building action head. ASP based on the large ruT5 model became the best model so far (4.0 points higher F1-score than baseline of the competition).

As some studies report [14], different coreference resolution models often do not transfer well to unseen domains. Moreover, for datasets containing news, such as RuCoCo, situations often arise when

[1] https://github.com/explosion/projects/tree/v3/experimental/coref
[2] https://github.com/yuvalkirstain/s2e-coref
[3] https://github.com/lyutyuh/ASP
[4] https://huggingface.co/cointegrated/LaBSE-en-ru
[5] https://huggingface.co/ai-forever/ruRoberta-large

new words, concepts and entities are encountered in test split. Probably the only way to overcome this is to increase or augment the dataset. This will likely lead to improved generalization, better handling of rare words and phrases, reduction of overfitting, improved robustness, because with more training data, the model is exposed to a greater variety of language patterns, more instances of rare words and phrases, more diverse examples, such as different writing styles, genres, or domains. However, it is important to consider the quality of the data. Therefore, while training models we use loss, which is weighted according to the data quality. We distinguish three classes of datasets: "gold", "silver" and "bronze". The lower the data quality, the lower the weight. The following three sections provide information about the ways that we used to increase the dataset.

3.1 Adding More Russian Coreference Resolution Datasets

To increase the size of the dataset we used two previously known good quality coreference resolution datasets in Russian:

- RuCor [15] – 163 documents;
- AnCor [16] – 521 document.

These datasets are considered "gold" along with RuCoCo.

3.2 Translating OntoNotes from English

OntoNotes 5.0 is one of the most popular datasets for coreference resolution in English with high quality. In some of our experiments with multilanguage models (more specifically – models with Russian and English tokens) we used this dataset directly as "silver" training data. But our final model was Russian only, thus we used the translation as "bronze" dataset.

To accurately translate the dataset into Russian, we did the following:

1. Use Helsinki-NLP/opus-mt-en-ru[6] for machine translation;
2. Translate the sentence and its clusters with entity spans to Russian language;
3. For every translated entity span find the most similar part of translated sentence using sentence encoder for text similarity (we used cointegrated/LaBSE-en-ru);
4. Use that span in the sentence as proper translation of original entity.

This may not be the most efficient approach, but it helped to achieve a translation with about 3% of all entities lost (1.68 entities lost per entire document in average). After analysing the results, we were satisfied with such a translation. We got 3017 new documents.

3.3 Using Pseudo-Labelling

The last part of final dataset was gathered with pseudo-labelling. We considered texts from same and different domains, collected from Taiga or Web:

- Arzamas[7] (Fiction) – 140 documents;
- collection5[8] (News articles with manual PER, LOC, ORG markup) – 355 documents;
- Interfax[9] (News) – 638 documents;
- KP[10] (News) – 355 documents;
- Lenta[11] (News) – 602 documents;
- N+1[12] (News) – 538 documents;
- Plaintext Wikipedia dump 2018 (ru.txt.gz)[13] – 1550 documents.

[6] https://huggingface.co/Helsinki-NLP/opus-mt-en-ru
[7] https://linghub.ru/static/Taiga/Arzamas.zip
[8] http://www.labinform.ru/pub/named_entities/collection5.zip
[9] https://linghub.ru/static/Taiga/Interfax.rar
[10] https://linghub.ru/static/Taiga/KP.rar
[11] https://linghub.ru/static/Taiga/Lenta.rar
[12] https://linghub.ru/static/Taiga/nplus1.rar
[13] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-2735

By this point, the ASP model based on ai-forever/ruT5-large (former sberbank-ai/ruT5-large)[14] trained on RuCoCo was the best model that we had. It was used to produce labels, i.e. clusters of spans of entities for texts that we collected.

Initially, more documents were collected for each dataset, however, they were randomly selected in such a way that the distribution of text lengths was similar to that of the RuCoCo dataset. When pseudo-labelling procedure was done, all datasets were filtered in such a way, that entity count, cluster count, entities per text length and clusters per text length distributions were roughly similar to those of the RuCoCo dataset (Fig. 1, Fig. 2).

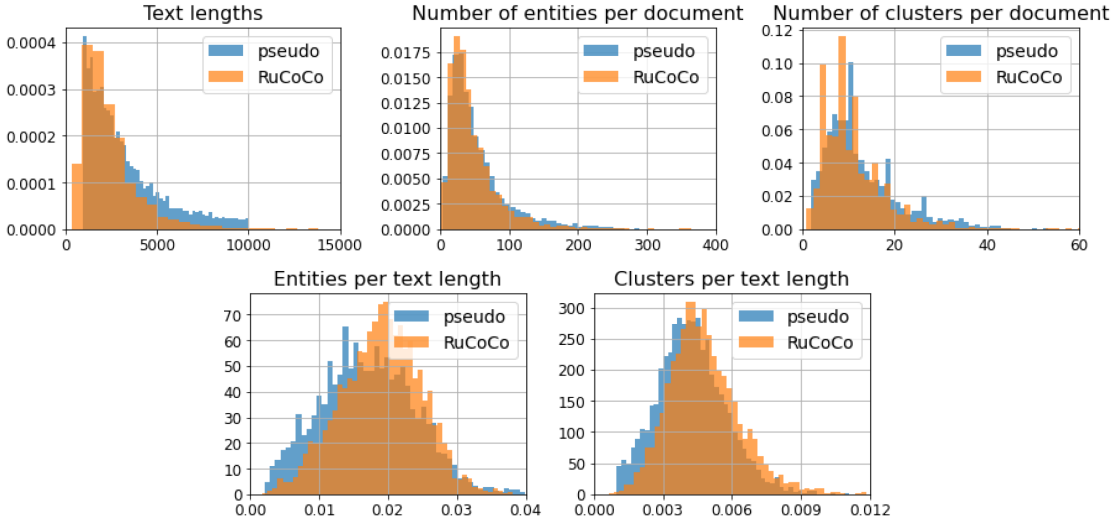

Figure 1: Comparative normalized histograms of two datasets: pseudo-labelled one and preprocessed RuCoCo

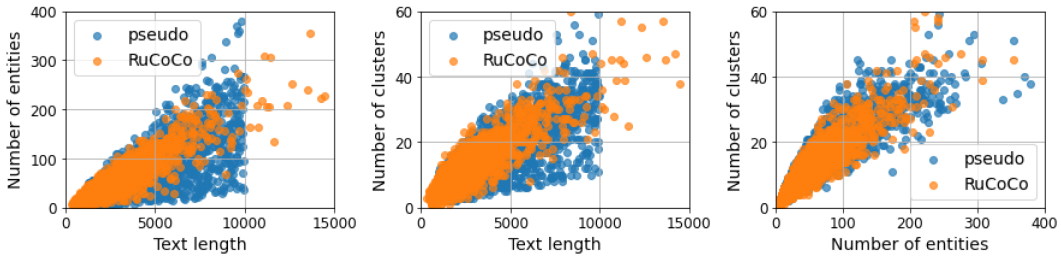

Figure 2: Scatterplots of different features of two datasets: pseudo-labelled one and preprocessed RuCoCo

A total of 3971 selected and labelled documents passed filtering and post processing. Final dataset is described in Table 1. OntoNotes Eng dataset was not used for final model. RuCoCo dataset is splitted into three sets – train, development and test (RuCoCo train split, RuCoCo dev split, RuCoCo test split, respectively) for local evaluation. All datasets together except OntoNotes Eng, RuCoCo dev split and RuCoCo test split are called “extra data” later in the paper.

[14] https://huggingface.co/ai-forever/ruT5-large

Dataset part	Number of documents	Dataset class	Loss weight
RuCoCo train split	2775	gold	1.0
RuCoCo dev split	150	gold	1.0
RuCoCo test split	150	gold	1.0
RuCor	163	gold	1.0
AnCor	521	gold	1.0
OntoNotes Rus	3017	bronze	0.1
Pseudo-labelled	3971	bronze	0.1
OntoNotes Eng[15]	3493	silver	0.5

Table 1: Final dataset parts and their sizes

4 Results and Analysis

For our final setup we used ASP based on ai-forever/ruT5-large utilizing transformers library [17]. We trained this model with input sequence length equal to 1550 tokens, hidden size of ASP action head equal to 4096, batch size equal to 1 for 18 epochs, which took 16 hours on a single nVidia RTX 3090Ti. Final dataset contained 10543 documents, including RuCoCo test split.

In Table 2 we present some results of different setups that we used during the competition. For evaluation we used LEA [18] as main metric for this competition. The last entry in bold is a result of the best model on private test of the competition.

Model	dev F1	test F1	leaderboard F1
baseline + cointegrated/LaBSE-en-ru	0.688	-	0.650
baseline + ai-forever/ruRoberta-large[16]	0.711	-	0.684
spacy-coref + cointegrated/LaBSE-en-ru	0.758	-	0.708
asp + cointegrated/rut5-base[17]	0.741	0.628	0.684
asp + cointegrated/rut5-base-multitask[18]	0.750	0.643	0.698
asp + ai-forever/ruT5-base[19]	0.765	0.650	0.699
asp + ai-forever/ruT5-large	0.791	0.664	0.727
asp + ai-forever/ruT5-large + extra data	0.786	0.667	0.733
asp + ai-forever/ruT5-large + extra data, test split, finetuned	0.799	-	0.738
asp, ai-forever/ruT5-large, extra data, test split, finetuned	**0.799**	**-**	**0.751**

Table 2: Evaluation results of different tested models

None of our models took into account split antecedents. That had an effect on recall metric. We tried to apply some simple models to handle this problem, and these models successfully increased recall, but all at cost of precision. Ultimately we could not achieve F1-score increase by handling split antecedents.

5 Ablation Study

Table 3 describes other experiments that included different base models and training techniques for ASP:

- Model 1 – final best model, added for comparison;
- Model 2 – one of the latest checkpoints of Model 1, but further trained a couple of epochs with only “gold” dataset, pseudo-labelled and translated data is excluded;
- Model 3 – google/mt5-large[20] model, but only with Russian and English tokens in dictionary, which is trained using entire available dataset, i.e. Model 1 dataset and OntoNotes Eng combined.

[15] https://huggingface.co/datasets/conll2012_ontonotesv5
[16] https://huggingface.co/ai-forever/ruRoberta-large
[17] https://huggingface.co/cointegrated/rut5-base
[18] https://huggingface.co/cointegrated/rut5-base-multitask
[19] https://huggingface.co/ai-forever/ruT5-base
[20] https://huggingface.co/google/mt5-large

Model	Dataset size	Training Epochs	Precision	Recall	F1	Private F1
Model 3	15584	4	0.7876	0.7972	0.7924	0.741
Model 2	11265 / 3797	13 / 5	0.7955	0.8083	0.8018	0.750
Model 1	11265	18	0.7925	0.8045	0.7985	0.751

Table 3: Evaluation results of top 3 final models on local development and global private split of RuCoCo. Model 2 was trained in two stages hence the separation in some columns. Dataset size is the number of documents after data preprocessing and it might be different with the initial number of documents in dataset because of long texts split

Model 3 is clearly undertrained and further experiments might bring some positive results. In addition, one can finetune Model 3 on some known tasks with sufficient multilanguage data before training it for coreference resolution.

Another experiments (Table 4) concerned the contribution of various dataset parts to the final result. Here we used ASP with cointegrated/rut5-base-multitask. RuCoCo train split was always a part of training data. Pseudo-labelled data was the same, i.e. acquired with ASP based on ai-forever/ruT5-large trained on RuCoCo.

Added data	dev split			test split			public test		
	P	R	F1	P	R	F1	P	R	F1
No added data	0.7468	0.7528	0.7498	0.753	0.561	0.643	0.739	0.652	0.693
RuCor, AnCor	0.7356	0.7594	0.7473	0.747	0.563	0.642	0.746	0.664	0.702
ONR	0.7417	0.7423	0.7420	0.753	0.544	0.632	-	-	-
PL	0.7589	**0.7895**	**0.7739**	0.748	0.577	0.652	-	-	-
ONR + PL	0.7431	0.7755	0.7590	0.735	**0.581**	0.649	-	-	-
ONE + ONR	0.7469	0.7595	0.7532	0.748	0.562	0.642	0.733	0.665	0.698
All data	**0.7658**	0.7657	0.7658	**0.765**	0.569	**0.653**	0.764	0.686	0.723

Table 4: Evaluation results with same model but different data. ONR – OntoNotes Rus, ONE – OntoNotes Eng, PL – Pseudo-Labelled. “All data” contains all unique datasets above in the table. Public test is what we managed to get while the development phase of the competition was active

6 Conclusions and Future Work

In this paper we present the research of approaches for coreference resolution in Russian language, results and details of our solution for Dialogue Evaluation 2023 RuCoCo competition. Our experiments reveal that the ASP model based on ai-forever/ruT5-large outperforms other coreference resolution models for Russian language, with the use of diverse and expanded training data, including translated OntoNotes and pseudo-labelled data, which significantly contributes to the model's performance. Our solution managed to take the first place in the competition. However, its performance still has room for improvement.

Future work should focus on exploring methods to handle split antecedents effectively. This is the most promising way to improve F1-score for such a task. Another aspect that our study highlights is the importance of diverse training data for model performance improvement. Training on pseudo-labelled data can be effective with small datasets within complex tasks. This technique also needs to be studied more precisely, since there are more ways to apply loss weighting and more data within different domains can be used. And last but not least, other backbone language models are applicable to this problem. One can use a multilanguage model with needed languages only [19] as a base transformer in ASP to more efficiently use datasets in another language, thus increasing the amount and diversity of training data even further.

Acknowledgements

The authors are grateful to colleagues at NTR Labs Machine Learning Research group for the discussions and support and to Prof. Sergey Orlov and Prof. Oleg Zmeev for the computing facilities provided.

References

[1] Brack A. et al. Coreference resolution in research papers from multiple domains //Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part I 43. – Springer International Publishing, 2021. – pp. 79–97.

[2] Liu Z., Shi K., Chen N. F. Coreference-aware dialogue summarization //arXiv preprint arXiv:2106.08556. – 2021.

[3] Morton T. S. Using coreference for question answering //Coreference and Its Applications. – 1999

[4] Kobayashi H., Malon C. Analyzing Coreference and Bridging in Product Reviews //Proceedings of the Fourth Workshop on Computational Models of Reference, Anaphora and Coreference. – 2022. – pp. 22–30.

[5] Stojanovski D., Fraser A. Coreference and coherence in neural machine translation: A study using oracle experiments //Proceedings of the Third Conference on Machine Translation: Research Papers. – 2018. – pp. 49–60.

[6] Lee K., He L., Zettlemoyer L. Higher-order coreference resolution with coarse-to-fine inference //arXiv preprint arXiv:1804.05392. – 2018.

[7] Kirstain Y., Ram O., Levy O. Coreference resolution without span representations //arXiv preprint arXiv:2101.00434. – 2021.

[8] Dobrovolskii V. Word-level coreference resolution //arXiv preprint arXiv:2109.04127. – 2021.

[9] Bohnet B., Alberti C., Collins M. Coreference Resolution through a seq2seq Transition-Based System //Transactions of the Association for Computational Linguistics. – 2023. – T. 11. – pp. 212–226.

[10] Liu T. et al. Autoregressive Structured Prediction with Language Models //arXiv preprint arXiv:2210.14698. – 2022.

[11] Liu T. et al. A structured span selector. // Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 2629–2641, Seattle, United States. Association for Computational Linguistics.

[12] Lee K., et al. End-to-end neural coreference resolution. //Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

[13] Beltagy I., Peters M. E., Cohan A. Longformer: The long-document transformer //arXiv preprint arXiv:2004.05150. – 2020.

[14] Toshniwal S. et al. On generalization in coreference resolution //arXiv preprint arXiv:2109.09667. – 2021.

[15] Ju T. S. et al. RU-EVAL-2014: Evaluating anaphora and coreference resolution for Russian //Komp'juternaja Lingvistika i Intellektual'nye Tehnologii. – 2014. – pp. 681–694.

[16] Budnikov A. E. et al. Ru-eval-2019: Evaluating anaphora and coreference resolution for russian //Computational Linguistics and Intellectual Technologies-Supplementary Volume. – 2019.

[17] Wolf T. et al. Transformers: State-of-the-art natural language processing //Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations. – 2020. – pp. 38–45.

[18] Moosavi N. S., Strube M. Which coreference evaluation metric do you trust? a proposal for a link-based entity aware metric //Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). – 2016. – pp. 632-642.

[19] David Dale. How to adapt a multilingual T5 model for a single language. URL: https://towardsdatascience.com/how-to-adapt-a-multilingual-t5-model-for-a-single-language-b9f94f3d9c90