





RuCAM: Comparative Argumentative Machine for the Russian Language

Maria Maslova¹, Stefan Rebrikov¹, Anton Artsishevski¹, Sebastian Zaczek²,
Chris Biemann² , and Irina Nikishina²  

¹ HSE University, Moscow, Russia

{mdmaslova,sarebrikov}@edu.hse.ru

² Universität Hamburg, Hamburg, Germany

{sebastian.zaczek,chris.biemann,irina.nikishina}@uni-hamburg.de

Abstract. Comparative question answering is one of the question answering subtasks which requires not only to choose between two (or more) objects, but also to explain the choice and support it with arguments. ChatGPT-like models are able nowadays to generate a coherent answer in a natural language, however, they are not fully reliable as they are not publicly accessible and tend to hallucinate. Another solution is a Comparative Argument Machine (CAM), which however, has been developed for English only. In this paper, we describe the development of RuCAM—comparative argumentative machine for Russian, as well as the challenges of the system adaptation for another language. It is the first open-domain system to argumentatively compare objects in Russian with respect to information extracted from the OSCAR corpus. We also introduce several datasets for the RuCAM subtasks: comparative question classification, object and aspect identification, comparative sentences classification. We provide models for each subtask and compare them with the existing baselines.

Keywords: comparative question answering · Russian language · comparative question identification · question answering

1 Introduction

The problem of choice has always been topical for people. In everyday life one may choose between different options of food or drink, various types of clothing. Besides, sometimes it is important to choose the right university, convenient smartphone or suitable operating system. While comparing several objects, people mainly look for substantiated answers why one item may be better than the other on a certain aspect. Hence, a possible NLP solution of this problem lies concurrently in the field of question answering and argument mining.

In this paper, we narrow our research to the specific type of questions—comparative questions, where two objects are compared with each other, optionally, by some aspect. For example, in the sentence: “*Who is a better friend, a cat*

or a dog?”,—“cat” and “dog” are the objects being compared and “better friend” is the aspect by which both objects are compared. The comparative question answering task aims at answering this question not only by choosing the winning object, but also by explaining and supporting the decision with arguments. Comparative questions with more objects and superlative questions like “Who is the tallest man in the world” are out of scope of our research.

The complexity of comparative question answering task makes it hard to find a universal instrument meeting all users’ requests. Specific product comparison systems, such as Compare.com¹, provide detailed information about a narrow range of goods (electric vehicles, certain hotels etc). Question answering platforms like Quora, or StackExchange² contain few comparative question answers regarding the total number of topics. Modern services based on Large Language Models (LLMs) are also able to provide a comparison of objects. However, they cannot be fully reliable as they are either not fully open (e.g. ChatGPT, Bing AI search) or they might hallucinate [7, 10] or the provenance of the arguments in such models cannot be derived.

One of the most known and prominent research in the field is Comparative Argumentative Machine (CAM) [3, 21] based on argumentative structures extracted from web-scale text resources. It allows to retrieve and rank textual argumentative structures relevant to a comparative user input of two objects. However, it is English-oriented, whereas there are no analogues for Russian.

Therefore, in this paper we present RuCAM—a system aimed at comparing two objects from general domain in Russian with argumentative explanations. We also provide the link to the web service³ and to the GitHub repository⁴ with code and data for all steps. As compared with its predecessor, CAM, it has the following differences: (i) it allows to work with comparative questions in natural language; (ii) it has the component for object and aspect identification from comparative questions; (iii) it uses an Elasticsearch index of Open Super-large Crawled Aggregated coRpus (OSCAR) [18]. The system is aimed to help users to speed up the process of comparative answers search. Moreover, a summary provided in the answer serves to support a decision-making process.

In this paper, we do not only describe a similar system and a pipeline from the engineering perspective. We also pose the following research questions: **(RQ1)** What are the main peculiarities of CAM that need to be taken into account when adapting it to other languages? **(RQ2)** What are the main challenges while adopting CAM specifically to the Russian language? In addition to the answers at those RQs, we also present the following contributions of the research:

1. We present RuCAM—a system for argumentative comparison of two objects based on information extracted from Common Crawl. As a successor of CAM, it is adapted for the peculiarities of the Russian language, has additional components that allow to work with natural questions and applies more recent NLP models.

¹ <https://compare.com>.

² <https://quora.com>, <https://stackexchange.com>.

³ <https://rucam.ltdemos.informatik.uni-hamburg.de>.

⁴ https://github.com/stefanrer/MCQA_RUS.

2. We provide several datasets for the RuCAM subtasks: comparative question classification, object and aspect identification in comparative questions, comparative sentence classification on general domains.
3. We provide baselines for each subtask in the pipeline and compare model performance with the existing approaches, if available.

2 Related Work

Comparative question answering derives from two tasks in Natural Language Processing (NLP): question answering and argument mining. There are multiple works that considered each problem separately, whereas very few studies combine both of them. In this section, we quickly overview each of the above mentioned tasks and then discuss the CAM system as the predecessor of RuCAM.

Recent studies on Comparative question answering [2] also deal with several problems in the field: comparative question classification, object and aspect identification and sentence classification. The identification of these elements will help to identify the comparative character of a question and its stance. These subtasks have been performed with the help of manually annotated question dataset and different neural networks, among which the fine-tuned RoBERTa has shown the best results. The extension of the above idea is described by Chekalina et al. [8]. Natural Language Understanding module is added to the argument search engine as well as Answer Generation module. The whole system is able to process a user’s request, to wit, extract objects, aspects and predicates, find the pros and cons argumentation for the objects found, and generate a short answer which is a summary of arguments for the English language.

As for the Russian language, some research studies on question classification have been conducted in recent years. One of the deepest analysis of comparative questions for Russian as well as developing methods for their classification has been done by [3]. The authors introduce 10 subclasses of comparative questions and claim to collect 50,000 questions and analyse 6,250 questions in Russian, however, they are not allowed to disclose the data even for academical purposes. In this paper, we compare our model with the model developed by [3] and make the data public. Other papers devoted to the comparative questions in Russia, are [16, 17], where the authors apply regular expressions, machine learning and neural networks methods to classify questions, including comparative ones, with respect to the predetermined typology.

Considering previous research on Argument Mining, there exist multiple publications on the topic for the Russian language. The most popular dataset and a variety of classification tasks are presented in [14], which describes different methods participated in the organised shared task. In [12], the authors present the first publicly available argument-annotated corpus of Russian based on Argumentative Microtext Corpus and experiment with feature-based machine learning approaches for argument identification. They also explore the possibility to classify argumentative discourse units (ADU) using traditional machine learning and deep learning methods [11]. More recent publications for the Russian language tackle the problem of argument generation given aspect [13] and end-to-end Argument Mining over varying rhetorical structures [9].

2.1 Comparative Argumentative Machine

As we stick to the concept of CAM search engine, it is necessary to point out its features that might be challenging to implement to other languages. Comparative Argumentative Machine [21] is aimed to help with answering comparative questions using argument mining techniques. It consists of 5 components: sentence retrieval, sentence classification, sentence ranking, aspect retrieval and result presentation.

First of all, CAM has no request processing step, whereas it is a significant part of RuCAM (Subsects. 3.1, 3.2). Objects and aspects are entered manually by user via interface. The first CAM step is the retrieval of relevant sentences from the CommonCrawl corpus. It searches for indexes corresponding entered objects and aspects using ElasticSearch. The RuCAM search is performed similarly (see Subject. 3.3) and is based on a cleaned and a preprocessed version of CC—Open Super-large Crawled Aggregated coRpus (OSCAR) [18]. The main adaptation challenge at this step is the availability of tools for large-scale text preprocessing.

The second CAM step is the sentence classification. The detailed process for this subtask is described in [19]. In a series of experiments with classification models, XGBoost shows the best results. For this task, RuCAM applies a similar baseline, but uses the Transformer-based models for Russian. This step might be challenging in terms of collecting good-quality annotations for training.

The next steps are similar for CAM and RuCAM: sentence ranking using a combination of Elastic Search and classification scores, and additional aspect retrieval. The final step—displaying results is almost similar for both systems, however RuCAM allows users to choose between entering a comparative question in Russian or to enter objects and aspects manually (see Fig. 2 for details).

3 System Design

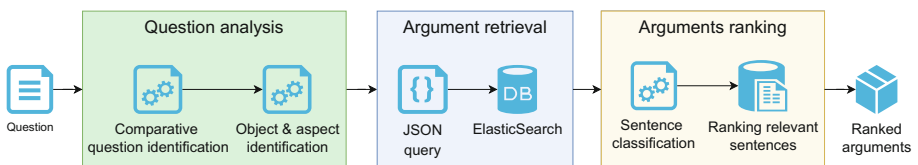


Fig. 1. Design of the RuCAM system.

As mentioned earlier, RuCAM is aimed at comparing two objects on the basis of argumentative structures extracted from web-scale text resources. This approach can be split in two consecutive steps: question analysis and argument retrieval. The first step is about identification of interrogative nature of sentence (is it comparative or not) as well as the sequence labelling subtask for object and aspect identification. The second step constitutes the search of argumentative structures relevant for objects extracted from the input question, their classification (“*Is the sentence in favour of one object or another?*”) and ranking (“*Which sentences are more relevant for comparison?*”).

The design of our RuCAM system is shown in Fig. 1. It consists of the following stages, which are described in more detail in the corresponding subsections: comparative question identification (Subsect. 3.1), object and aspect identification (Subsect. 3.2), argument retrieval (Subsect. 3.3), sentence classification (Subsect. 3.4), and sentence ranking (Subsect. 3.5).

3.1 Question Identification

The processing of the request starts with the identifying the question type (comparative or not). To do that, we first compile the dataset that satisfies our needs. We start with the Russian dataset of questions [16]: 146 items are used with positive tags (with classes “choice” and “comparison”) and all other sentences (2,121) as negative. As the major Russian dataset from [3] with comparative questions cannot be disclosed for training, we only take its open part used for testing the fine-grained classifier. Additionally, we translate query questions from [4–6] for argument mining Touché competitions using the “EN_RU” translation model from [15]. We manually check and improve comparative sentences in terms of fluency and grammaticality, as their number is relatively small.

However, the number of positive entries in the compiled dataset might be still too small to work with, therefore, we automatically translate the existing English dataset for comparative question identification [2, 3] with the model mentioned beforehand. The dataset used as well as their statistics and application is presented in Table 1. The example entries of the compiled Russian dataset could be seen in Table 2.

Table 1. Statistics and specifications of the datasets used for comparative question identification.

Dataset	Non-Comparative	Comparative	Total
Nikolaev et al. [16]	2,040	143	2,183
Bondarenko et al. (fine-grained test) [3]	0	1,240	1,240
Touché (2020–2022) (translated) [4–6]	0	100	100
Webis-CompQuestions-20 (translated) [3]	1,431	1,429	2,860
Webis-CompQuestions-22 (translated) [2]	2,529	3,088	5,617
Total (selected)	6,000	6,000	12,000

At this step, we implement and compare the following approaches for question identification: a rule-based baseline, a ru-tinyBERT model finetuned on the compiled dataset in Russian, and the fine-tuned BERT model from [3] (both trained on 3 epochs with batch 32, other parameters are default). We discuss the results and compare their performance in ‘Evaluation’ section. As a rule-based baseline we implement the idea of special patterns contained in questions as the simplest identification mechanism. Therefore, we created patterns which include comparative forms (“*лучше*” (“better”), “*хуже*” (“worse”)), explicit mentions of

comparison (“*по сравнению*” (“in comparison with”)), similarity (“*похожий*” (“similar”), “*одинаковый*” (“same as”)) or difference (“*отличается*” (“different from”), “*непохожий*” (“not like”)), advantages (“*преимущество*” (“advantages”), “*выигрывает*” (“wins over”)) or disadvantages (“*недостаток*” (“disadvantage”), “*проигрывает*” (“lags behind”)), verbs expressing choice (“*выбрать*” (“choose”), “*предпочесть*” (“prefer”)).

Table 2. Example entries from the Russian dataset for comparative question identification. 1 stands for the comparative sentence, 0 for non-comparative.

Sentence	Label
Каковы преимущества и недостатки PHP по сравнению с Python? <i>Kakovy preimushhestva i nedostatki PHP po sravneniyu s Python?</i> What are the advantages and disadvantages of PHP compared to Python?	1
Когда мне поступить в университет? <i>Kogda mne postupit' v universitet?</i> When should I go to university?	0

3.2 Object and Aspect Identification

After identifying the question as comparative, we need to extract objects and (optionally) aspects to further provide them for the argument retrieval stage. In order to create a dataset for the task, we take 6000 sentences from the previous step that were labelled as comparative and manually annotate them. Three experts in computational linguistics and NLP are required to highlight “*object-1*”, “*object-2*”, “*aspect*”, “*common object*” in a text, analogously to the guidelines in [1]. *Common object* is the specific structure with a noun subordinating two adjectives in a construction like “*черный или зеленый чай*” (“green or black tea”). The level of annotator agreement (Fleiss’s kappa) amounts to the following levels: *object-1* & *object-2*—0.83, *aspect*—0.71, *common object*—0.54. When creating the final dataset for models fine-tuning (2,328 sentences with exactly two objects), we use the annotation versions supported by the majority of annotators.

At this step, we also implement several approaches of object identification: a rule-based algorithm, fine-tuned Transformer encoders (10 epochs, batch 32, other parameters are default) and a few-shot approach on generative Transformers. A rule-based algorithm is founded on the idea that all requests have a certain structure. By this, we mean the existence of two comparison objects and a connective between them. We consider the following cases: two nouns, two verbs, combinations of noun and adjective, combinations of noun and two subordinate adjectives. We also expect a connective from the list of conjunctions and synthetic words expressing comparison between the objects: “*или*” (“or”), “*лучше*” (“better”), “*лучше чем*” (“better than”), “*лучше ADV чем SCONJ*” (“better than”), “*лучше А чем CONJ по PR сравнению NOUN*” (“better than in comparison with”).

3.3 Sentence Retrieval

In order to retrieve arguments in favour of one or the other object, we use Open Super-large Crawled Aggregated coRpus (OSCAR) [18] which comprises 21.5M documents for the Russian language which we split into 21B sentences. We use OSCAR instead of the Common Crawl, while it is claimed to be its filtered version. We store and index this data with Elasticsearch⁵—a search engine based on the Lucene library that enables fast search using HTTP web interface and schema-free JSON. To ensure quick and stable responses from Elasticsearch we deploy it in parallel on 16 Ubuntu Server 16.04 nodes with 2×10 Cores (+Hyper-threading) CPU, 256 GB of RAM and 4TB HDD each.

When indexing documents, we decide to create two indexes: the first one is used for storing document information (the number of sentences in the document, its metadata and the web-link) and the second one for storing sentences. Each indexed sentence includes its document ID, previous and next sentence IDs, number of words in the sentence and the text itself.

To retrieve sentences, we first do the Snowball stemming of the query objects and aspects and then apply wildcards to be able to find all word forms. Lemmatization could be a fair alternative at this step, however, we stick to stemming because of the time constraints. We send a boolean json query and require that the clause must appear in matching documents. We consider this step to be the most challenging in the whole CAM for implementation, as Russian language has a highly fusional morphology which makes it much more difficult to retrieve sentences than in English, as query words may occur in any form.

3.4 Sentence Classification

Table 3. Examples with tags from the sentence classification dataset. Objects for comparison are in **bold**.

Sentence	Tag
В любом случае, запекать гораздо полезнее, чем жарить <i>V ljubom sluchae, zapekat' gorazdo poleznee, chem zharit'</i> In any case, baking is much healthier than frying	BETTER
Тащить лыжи на гору сложнее, чем один сноуборд <i>Tashhit' lyzhy na goru slozhnee, chem odin snoubord</i> Carrying ski up the mountain is harder than just one snowboard	WORSE
Здесь лучше всего использовать спонж или широкую кисть <i>Zdes' luchshe vsego ispol'zovat' sponzh ili shirokuyu kist'</i> Here it is best to use a sponge or a wide brush	NONE

After the candidate sentences with possible arguments are found, it is necessary to understand whether a sentence argues in favour of the first or the second

⁵ <https://www.elastic.co>.

object. Analogously to CAM [21], we collect a dataset of 1208 sentences from 67 object pairs and annotate them using Yandex.Toloka system for data crowdsourcing [20]. To do this, we select same or similar pairs from the same domains as in English (e.g., programming languages, car manufacturers, food and drinks) and make a query to Elasticsearch as it is described in Sect. 3.3 to extract all sentences matching the query. Then we create three tags: “*BETTER*” (the first item is better or “wins”)/“*WORSE*” (the first item is worse or “loses”) or “*NONE*” (the sentence does not contain a comparison of the target items). When displaying classified sentences, “*BETTER*”-sentences support the first compared object, “*WORSE*”-sentences are used as pro-argument for the second object. Unfortunately, the annotated dataset is highly imbalanced: 75% of texts belong to the “*NONE*” tag, 16%—to the “*BETTER*” tag and only 9%—to the “*WORSE*” tag. Table 3 demonstrates some excerpts from the dataset within the label.

At this step, we also implement a rule-based baseline, several large language model classifiers based on Transformer Encoders (3 epochs, batch 32, other parameters are default) and few-shot approaches with generative Transformers, which allow to address in issue of data imbalance using only 5 examples from each class. The rule-based approach requires two lists of keywords with adjectives and adverbs with the meaning of superiority or inferiority of the first object over the second. We also take into account negation cases when the sense of a sentence is reversed.

3.5 Sentence Ranking and Object Comparison

The processes of sentence ranking and object comparison is identical to the one in CAM [21]: we score comparative sentences by combining the classifier confidence and the Elasticsearch score⁶. When displaying the arguments in RuCAM on a certain object, we sum up not only *BETTER*-arguments, where the current object is the first item, but also *WORSE*-arguments, where the object is the second item in the sentence. For instance, both sentences “Python лучше, чем Java” (class $>$) and “Java хуже, чем Python” (class $<$) are used in favour of Python when comparing with Java.

In addition to user-specified comparison aspects, CAM generates up to ten supplementary aspects (even when no comparison aspect at all was provided by the user) to display it for better output presentation. To do the same for RuCAM, we use three different methods for aspect mining: (1) searching for comparative adjectives and adverbs; (2) searching for phrases with comparative adjectives/adverbs and a preposition like “для” (“for”), “чтобы” (“to”), etc. (e.g., “быстрее для написания кода” (“better for code writing”)); (3) searching for specific hand-crafted patterns like “из-за более высокой скорости” (“due to higher speed”), or “причина этого кроется в цене” (“the reason for this lies in the price”). An extracted aspect is assigned to the object with the higher co-occurrence frequency.

⁶ <https://www.elastic.co/guide/en/elasticsearch/guide/current/scoring-theory.html>.

4 Evaluation

This section is devoted to the evaluation of all of the developed models for each of the pipeline steps of RuCAM. We present comparison tables and select the best-performed model from each step to the final pipeline.

4.1 Question Identification

Table 4. Model comparison table for the results in the Comparative Question Classification task.

Model name	Model Parameters	Precision	Recall	F1-score
rule-based	–	0.89	0.88	0.87
Bondarenko et al. [3]	167.3M	0.95	0.95	0.95
ruBERT-tiny ^a	29.4M	0.91	0.90	0.90

^a<https://huggingface.co/cointegrated/rubert-tiny2>

From the results, presented in Table 4, we can see comparative questions are indeed very specific kind of questions and that they can be easily identified even with rule-based approaches. The best results are achieved by the existing model from [3], however, the questions the model was trained on are from the same dataset we took the major part of our testing questions from. This partially explains higher results of [3], as they were training on the data from the same distribution as the test set. Nevertheless, even with no access to this data, but only to machine-translated datasets, it is possible to train a well-performing model. The finetuned ruBERT-tiny achieves decent results outperforming the baseline with smaller number of parameters.

4.2 Object and Aspect Identification

This subsection presents the results for the developed token classification models for identifying objects and aspects. We test a rule-based approach, and several Transformer approaches: finetuning of the standard sequence taggers as well as testing few-shot generative Transformers. Table 5 presents the results for each model. As this subtask is formulated for the first time for the Russian language, there are no additional models to compare with.

Even though the rule-based model demonstrates quite high results, it still lags behind most of the provided LLMs. However, the results of this approach are still higher than most of the models on “*CommonObject*” identification. Generally, the results show that generative models perform on par or even slightly better than the baseline and significantly lag behind Transformer encoders. Nevertheless, we need to specify that generative Transformers might have higher potential as they were shown 5 examples only and they might perform much better after

Table 5. Results on F1-score for the object and aspect identification experiments.

Model name	Object-1	Object-2	Aspect	Common Object	Average
rule-based baseline	0.71	0.81	0.17	0.23	0.66
wikineural-multilingual-ner ^a	0.77	0.75	0.50	0.66	0.71
ruBERT-tiny ^b	0.87	0.78	0.17	0.00	0.69
ruBERT-large ^c	0.97	0.88	0.57	0.00	0.80
bactrian-x-llama-13b ^d (5-shot)	0.75	0.72	0.22	0.00	0.62
ruGPT3 (5-shot)	0.66	0.65	0.04	0.00	0.54

^a<https://huggingface.co/Babelscape/wikineural-multilingual-ner>

^b<https://huggingface.co/cointegrated/rubert-tiny>

^c<https://huggingface.co/ai-forever/ruBert-large>

^d<https://github.com/mbzuai-nlp/bactrian-x>

proper finetuning. We leave this question out of scope of our research and present models as baselines only. Regarding lower and zero scores for the “*Aspect*” and “*CommonObject*” labels for many models, we assume that the reason of that is the inconsistency in annotations. Similarly low results we also shown in [1], which might also indicate the difficulty of the “*Aspect*” and “*CommonObject*” identification in general.

4.3 Sentence Classification

Table 6. The results on F1-score for the comparative sentence classification models.

Model	BETTER	WORSE	NONE	Average
rule-based	0.34	0.33	0.82	0.69
ruBERT-tiny ^a	0.57	0.38	0.91	0.82
ruBERT-large ^b	0.00	0.00	0.87	0.76
bactrian-x-llama-13b (5 × 3-shot) ^c	0.30	0.12	0.32	0.36
ruGPT3 (5 × 3-shot)	0.26	0.24	0.19	0.23

^a<https://huggingface.co/cointegrated/rubert-tiny>

^b<https://huggingface.co/ai-forever/ruBert-large>

^c<https://github.com/mbzuai-nlp/bactrian-x>

As in the previous steps, we compare a rule-based and several Transformer-based approaches. According to Table 6, the results for comparative sentence classification are inconsistent for each class and relatively low for all of the presented models, due to the class imbalance problem. The “*WORSE*” class always achieves the lowest score among the classes. We can see that the best results are achieved with the ruBERT-tiny model, while BERT-large overfits on the dataset with prevalent “*NONE*” class. Rule-based approaches also produce average results for all classes while the lowest scores are achieved by LLMs (ruGPT3 and bactrian-x-llama) with generative setup.

5 Demonstration System and Current Work

The main outcome of our research is the final system where we integrate all the parts described above. Figure 2 depicts the interface of the whole system. We have decided to apply ruBERT-tiny at each step as the compromise between speed, memory space and efficiency. The evaluation of the system is currently work in progress. We plan to evaluate RuCAM analogously to CAM evaluation pipeline, by asking whether users are faster answering correctly when using the CAM system and ask some users to “play” to collect some user experience feedback. The research is to be based on the collection of topics (two objects + one aspect) available for CAM- and keyword-based search. A topic is suitable for the research if it has more support sentences than the established lower bound. Additional descriptions fore some topics will help to avoid potential ambiguities or subjectivities. In the first part of the task the participants should give an answer as quickly as possible using both experimental systems alternatingly. For that purpose the collection of topics is randomly split into two groups. The second part of the task allows users to test the functionality and convenience of the system without time limitations and make as much comparisons as they want.

Comparative Argumentative Machine for Russian (RuCAM)

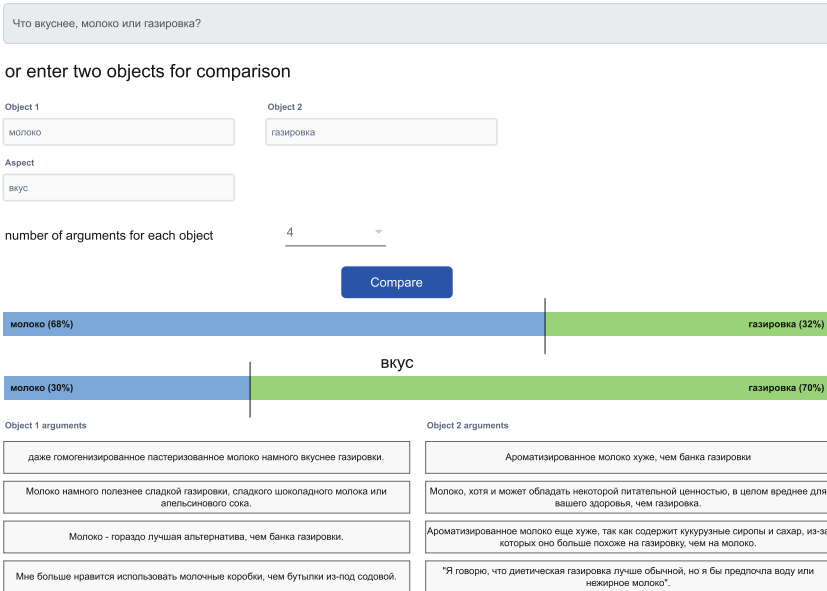


Fig. 2. Design interface of the RuCAM demonstration system.

6 Conclusion

In this article, we present RuCAM—the first instrument which helps to answer general-domain comparative questions in Russian. Inspired by the CAM system, we create a similar pipeline, adding new steps for comparative question classification, object and aspect identification, sentence classification. We also present several new datasets in Russian that might be further used for the fine-tuning of language models for each subtask. From the performed experiments, we can see that the rule-based approaches show decent results on all subtasks of comparative QA as well as few-shot generative Transformers (which need further investigation). As the answer on *(RQ1)*, we state that when transferring CAM to other languages, you should take the following peculiarities into account: (i) the difference in the notion of comparative sentences in different languages; (ii) the difference in the syntax and morphology of languages when re-implementing rule-based approaches; (iii) the existence of the relevant datasets and pre-trained Large Language Models for training and large text corpora containing comparative sentences for search in the target language. Nevertheless, as it has been shown for Russian, it might be quite smooth if at least some the required tools are available. As for *(RQ2)*, we can see that the main challenge is the complexity of Russian grammar for re-implementing rule-based approaches. Inflectional morphemes make it difficult to search for specific forms in the text at any step of our approach. Moreover, quite flexible word order which makes the process of matching regular expressions in the rule-based approaches more challenging. As future directions, we plan to incorporate a summarisation system that would be able to produce a coherent answer from two lists of arguments for each object. It will allow us to compare the results of various instruct-tuned models for Russian and ChatGPT with the RuCAM pipeline. As the instruct-tuned generative models are well-suited for such type of tasks, it would be a great study to understand in how many cases these models provide reasonable arguments, and how often they hallucinate in comparison to RuCAM. Another challenging direction is to apply on discourse analysis approaches to identify the argumentative sentences. Utilizing such methods may retrieve more coherent text spans.

Acknowledgements. This work was supported by the DFG through the project “ACQuA: Answering Comparative Questions with Arguments” (grants BI 1544/7- 1 and HA 5851/2- 1) as part of the priority program “RATIO: Robust Argumentation Machines” (SPP 1999).

References

1. Beloucif, M., Yimam, S.M., Stahlhacke, S., Biemann, C.: Elvis vs. M. Jackson: who has more albums? Classification and identification of elements in comparative questions. In: Calzolari, N., et al. (eds.) Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20–25 June 2022, pp. 3771–3779. European Language Resources Association (2022). <https://aclanthology.org/2022.lrec-1.402>

2. Bondarenko, A., Ajjour, Y., Dittmar, V., Homann, N., Braslavski, P., Hagen, M.: Towards understanding and answering comparative questions. In: Candan, K.S., Liu, H., Akoglu, L., Dong, X.L., Tang, J. (eds.) WSDM 2022: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event/Tempe, AZ, USA, 21–25 February 2022, pp. 66–74. ACM (2022). <https://doi.org/10.1145/3488560.3498534>
3. Bondarenko, A., et al.: Comparative web search questions. In: Caverlee, J., Hu, X.B., Lalmas, M., Wang, W. (eds.) WSDM 2020: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, 3–7 February 2020, pp. 52–60. ACM (2020). <https://doi.org/10.1145/3336191.3371848>
4. Bondarenko, A., et al.: Overview of Touché 2020: argument retrieval. In: Aramatzis, A., et al. (eds.) CLEF 2020. LNCS, vol. 12260, pp. 384–395. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58219-7_26
5. Bondarenko, A., et al.: Overview of touché 2022: argument retrieval. In: Faggioli, G., Ferro, N., Hanbury, A., Potthast, M. (eds.) Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, 5th–8th September 2022. CEUR Workshop Proceedings, vol. 3180, pp. 2867–2903. CEUR-WS.org (2022). <https://ceur-ws.org/Vol-3180/paper-247.pdf>
6. Bondarenko, A., et al.: Overview of Touché 2021: argument retrieval. In: Candan, K.S., et al. (eds.) CLEF 2021. LNCS, vol. 12880, pp. 450–467. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85251-1_28
7. Cao, M., Dong, Y., Cheung, J.: Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland (Volume 1: Long Papers), pp. 3340–3354. Association for Computational Linguistics (2022). <https://doi.org/10.18653/v1/2022.acl-long.236>
8. Chekalina, V., Bondarenko, A., Biemann, C., Beloucif, M., Logacheva, V., Panchenko, A.: Which is better for deep learning: Python or matlab? Answering comparative questions in natural language. In: Gkatzia, D., Seddah, D. (eds.) Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, EACL 2021, Online, 19–23 April 2021, pp. 302–311. Association for Computational Linguistics (2021). <https://doi.org/10.18653/v1/2021.eacl-demos.36>
9. Chistova, E.: End-to-end argument mining over varying rhetorical structures. In: Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, pp. 3376–3391. Association for Computational Linguistics (2023). <https://doi.org/10.18653/v1/2023.findings-acl.209>. <https://aclanthology.org/2023.findings-acl.209>
10. Dale, D., Voita, E., Barrault, L., Costa-jussà, M.R.: Detecting and mitigating hallucinations in machine translation: model internal workings alone do well, sentence similarity Even better. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Toronto, Canada (Volume 1: Long Papers), pp. 36–50. Association for Computational Linguistics (2023). <https://aclanthology.org/2023.acl-long.3>
11. Fishcheva, I., Goloviznina, V., Kotelnikov, E.V.: Traditional machine learning and deep learning models for argumentation mining in Russian texts. CoRR abs/2106.14438 (2021). <https://arxiv.org/abs/2106.14438>
12. Fishcheva, I., Kotelnikov, E.: Cross-lingual argumentation mining for Russian texts. In: van der Aalst, W.M.P., et al. (eds.) AIST 2019. LNCS, vol. 11832, pp. 134–144. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-37334-4_12

13. Goloviznina, V., Fishchev, I., Peskischeva, T., Kotelnikov, E.: Aspect-based argument generation in Russian. In: Computational Linguistics and Intellectual Technologies: Papers from the Annual Conference “Dialogue” (2023)
14. Kotelnikov, E.V., Loukachevitch, N.V., Nikishina, I., Panchenko, A.: RuArg-2022: argument mining evaluation. In: Computational Linguistics and Intellectual Technologies: Papers from the Annual Conference “Dialogue” (2022)
15. Ng, N., Yee, K., Baevski, A., Ott, M., Auli, M., Edunov, S.: Facebook FAIR’s WMT19 news translation task submission. In: Proceedings of the Fourth Conference on Machine Translation, Florence, Italy (Volume 2: Shared Task Papers, Day 1), pp. 314–319. Association for Computational Linguistics (2019). <https://doi.org/10.18653/v1/W19-5333>. <https://aclanthology.org/W19-5333>
16. Nikolaev, K., Malafeev, A.: Russian-language question classification: a new typology and first results. In: van der Aalst, W.M.P., et al. (eds.) AIST 2017. LNCS, vol. 10716, pp. 72–81. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73013-4_7
17. Nikolaev, K., Malafeev, A.: Russian Q&A method study: from Naive Bayes to convolutional neural networks. In: van der Aalst, W.M.P., et al. (eds.) AIST 2018. LNCS, vol. 11179, pp. 121–126. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-11027-7_12
18. Ortiz Suárez, P.J., Sagot, B., Romary, L.: Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. In: Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, pp. 9–16. Leibniz-Institut für Deutsche Sprache, Mannheim (2019). <https://doi.org/10.14618/ids-pub-9021>. <http://nbn-resolving.de/urn:nbn:de:bsz:mh39-90215>
19. Panchenko, A., Bondarenko, A., Franzek, M., Hagen, M., Biemann, C.: Categorizing comparative sentences. In: Stein, B., Wachsmuth, H. (eds.) Proceedings of the 6th Workshop on Argument Mining, ArgMining@ACL 2019, Florence, Italy, 1 August 2019, pp. 136–145. Association for Computational Linguistics (2019). <https://doi.org/10.18653/v1/w19-4516>
20. Pavlichenko, N., Stelmakh, I., Ustalov, D.: CrowdSpeech and VoxDIY: benchmark dataset for crowdsourced audio transcription. In: Vanschoren, J., Yeung, S. (eds.) NeurIPS Datasets and Benchmarks 2021, December 2021, virtual (2021)
21. Schildwächter, M., Bondarenko, A., Zenker, J., Hagen, M., Biemann, C., Panchenko, A.: Answering comparative questions: better than ten-blue-links? In: Azzopardi, L., Halvey, M., Ruthven, I., Joho, H., Murdock, V., Qvarfordt, P. (eds.) Proceedings of the 2019 Conference on Human Information Interaction and Retrieval, CHIIR 2019, Glasgow, Scotland, UK, 10–14 March 2019, pp. 361–365. ACM (2019). <https://doi.org/10.1145/3295750.3298916>