Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference "Dialogue 2025"

April 23–25, 2025

Structured sentiment analysis using few-shot prompting of an ensemble of LLMs

Petr Rossyaykin
Lomonosov Moscow State University
Moscow, Russia
petrrossyaykin@gmail.com

Abstract

This paper describes our participation in RuOpinionNE-2024 shared task (Loukachevitch et al., 2025). The objective of this task is to extract opinion tuples of the form <holder, target, polarity expression, polarity> from news texts in Russian. We approached this task with few-shot prompting of super large language models (LLMs). The quality of LLMs' predictions was improved in two ways. In the first stage we used prompts with examples which text embeddings were similar to that of the target text. In the second stage we augmented prompts with answers of LLMs from the previous stage, achieving the second-best F1 score in the competition in the post-evaluation stage. Our results show that the addition of answer suggestions to the prompt is particularly useful if they provide novel and variable information.

Keywords: structured sentiment analysis, opinion extraction, few-shot prompting, large language models
DOI: 10.28995/2075-7182-2025-23-291-298

Few-shot prompting ансамбля больших языковых моделей для структурированного анализа тональности

Аннотация

В этой статье мы описываем наше участие в соревновании RuOpinionNE-2024. Задача соревнования состоит в извлечении кортежей мнений вида <источник мнения; объект мнения; выражение, содержащие мнение; полярность> из русскоязычных новостных текстов. Для решения этой задачи мы использовали few-shot prompting больших языковых моделей. Мы использовали два способа улучшения качества их предсказаний. На первом этапе в качестве примеров в промпте мы использовали тексты, векторные представления которых близки к векторному представлению целевого текста. На втором этапе мы добавили к промпту предсказания моделей, полученные на предыдущем этапе. Таким образом, на стадии post-evaluation мы достигли второго результата соревнования по F-мере. Наши результаты показывают, что добавление к промпту предлагаемых вариантов ответа улучшает качество, если они содержат новую и разнообразную информацию.

Ключевые слова: структурированный анализ тональности, извлечение мнений, few-shot prompting, большие языковые модели

1 Introduction

Sentiment analysis has been one of the major tasks in NLP for decades (Pang et al., 2002; Turney, 2002; Wiebe et al., 2005; Liu, 2012; Nakov et al., 2013; Socher et al., 2013; Pontiki et al., 2014; Golubev et al., 2023). Any opinion expressed in a text is instantiated in a polar expression (*e*) and has a holder (*h*), a target (*t*) and a polarity (*p*), so one has to recognize these four entities in order to extract an opinion from a text (Liu, 2012). Structured Sentiment Analysis (SSA) is thus defined as a task of extracting all opinion tuples of the form <*h*, *t*, *e*, *p*> from a given text (Barnes et al., 2021, 3387).

SSA is especially relevant for news texts, because, unlike e.g. reviews, they often do not have a single text-level sentiment but contain different opinions with different holders and/or targets or no opinions at all (Wiebe et al., 2005, 166). Despite this, SSA in news texts remains a particularly understudied subtask

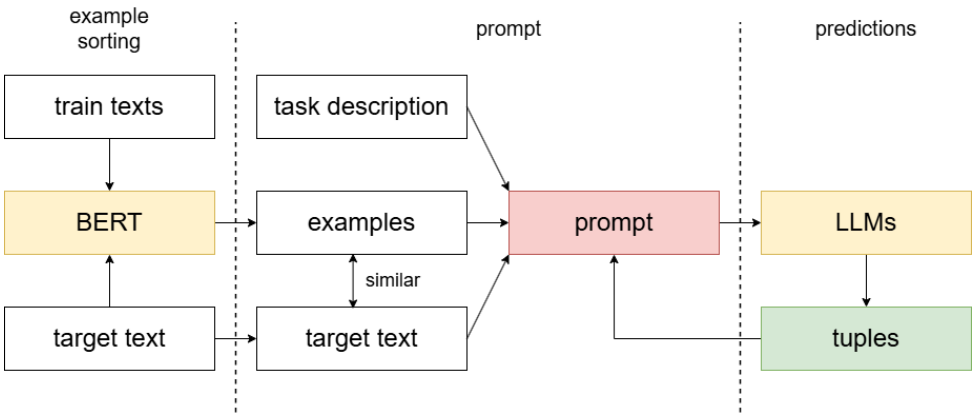


Figure 1: Overview of our approach.

of natural language understanding. For example, only one of seven datasets presented for shared task on SSA at SemEval 2022 contained news texts (Barnes et al., 2022, 1282). Incidentally, all but one of the participants of that task achieved the worst score on that particular dataset (Barnes et al., 2022, 1284), which suggests that news texts constitute a more difficult domain for SSA than reviews.

In this paper we discuss our participation in RuOpinionNE-2024, a shared task on extracting opinion tuples from Russian news texts (Loukachevitch et al., 2025). Our approach consisted in direct generation of opinion tuples using few-shot prompting of super large language models (LLMs). We used two tricks to improve the quality of LLMs' predictions. The first trick was to choose examples according to their semantic similarity to the target example. The second trick was to augment prompts with the answers from the first stage. This can be done cyclically and for any LLM both its own answers and the answers of other LLMs can be included. This allowed us to further improve the quality, although we found out that cyclic application of this procedure is not beneficial because LLMs benefit from novel and variable, rather than similar and recycled, information. Figure 1 schematically presents our method. We describe the method, the models used and the results in more detail in Sections 3–5.

2 Task description and the data

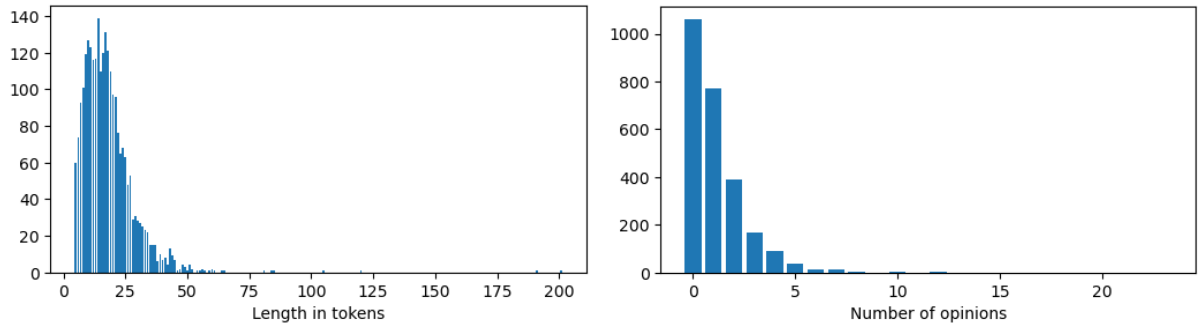
RuOpinionNE-2024 presents the task of opinion tuple extraction from Russian news texts. Opinion tuples have the following form: *<h(older)*, *t(arget)*, *e(xpression)*, *p(olarity)>* (the polarity is either POS(itive) or NEG(ative)). The train set of the competition consists of 2556 texts annotated for structured sentiment and the test set consists of 803 texts. The texts are mostly single sentences taken from news texts in Russian, although there are several texts which contain multi-sentential direct speech. The average length of texts in the train set is 18.1 tokens, i.e. orthographic words, the shortest text is 5 tokens long and the longest one consists of 201 tokens. The distribution of the lengths of texts in the train set is shown in Figure 2a.

Train set texts contain from 0 to 23 opinions, although only 5 texts contain more than 10 opinions and only 38 texts contain more than 5 opinions. 1062 texts contain no opinions. The number of opinions per text is shown in Figure 2b.

It is also of note that there are 2904 opinions in the train set in total and in 1281 cases the opinion holder is covert ('NULL' in 897 cases and 'AUTHOR' in 384 cases).

3 Few-shot prompting approach to SSA

The gist of our approach is to generate opinion tuples directly by prompting super large language models.



(a) Distribution of texts' lengths in the train set

(b) Number of opinions per text in the train set

Figure 2: Basic properties of the train set

3.1 The first stage: basic prompt

The basic prompt we used is given below in the left column with two examples ("shots") in the prompt. The English translation of the prompt is given in the right column.

Ты эксперт в оценке тональности.

Тебе нужно найти все негативные и позитивные отношения между сущностями в тексте и вывести их в следующем формате:

[источник отношения, объект отношения, выражение в тексте содержащее оценку, оценка (POS/NEG)]

Если источником отношения является автор, то пиши:

['AUTHOR', объект отношения, выражение в тексте содержащее оценку, оценка (POS/NEG)]

Если выраженного источника нет, то пиши:

['NULL', объект отношения, выражение в тексте содержащее оценку, оценка (POS/NEG)]

Допустимо вернуть пустой ответ:

[]

Не нужно давать пояснений к ответу.

Примеры

Текст: По итогам первого полугодия 2016 года банк занимает 41-е место по размеру активов.

Ответ: []

Текст: Русская Википедия в четвёртый раз получила «Премию Рунета»

Ответ: [['NULL', 'Русская Википедия', 'Премию Рунета', 'POS']]

Текст: (К слову четвертое место в списке лидеров по итогам 2010 года занимает Москва).

Ответ:

You are an expert in sentiment analysis.

You need to identify all positive and negative relations between entities in the text and present them in the following format:

[source of sentiment, target, polar expression, polarity (POS/NEG)]

If the author is the source of sentiment, write:

['AUTHOR', target, polar expression, polarity (POS/NEG)]

If there is no explicit source, write:

['NULL', target, polar expression, polarity (POS/NEG)]

Returning an empty answer is allowed:

[]

Do not provide any explanations in the response.

Examples

Text: As of the first half of 2016, the bank ranks 41st in terms of assets.

Answer: []

Text: The Russian Wikipedia has received the "Runet Prize" for the fourth time.

Answer: [['NULL', 'Russian Wikipedia', 'Runet Prize', 'POS']]

Text: (By the way, Moscow ranks fourth in the list of leaders based on the results of 2010.)

Answer:

This prompt consists of three parts: the instruction, “shots” and the target text, which is *(К слову четвертое место в списке лидеров по итогам 2010 года занимает Москва)* in the given example. We used only the “user” part of the prompt without splitting it into the “system” and the “user” parts, because this resulted in a slightly worse quality during our preliminary experiments.

The “shots” were chosen according to cosine similarity of their text embeddings to that of the target text. We used mean pooling of token embeddings from BERT output to get text embeddings. For each target sentence, we added *n* examples with most similar texts after the instruction, where *n* is the number of shots.

The opinion tuples generated by LLMs given prompts of this kind constituted the predictions in the first stage.

3.2 The second stage: augmented prompt

We augmented the prompt with predictions obtained in the first stage. They were added to the prompt after the list of examples. An example of augmented prompt is given below (left column) together with its translation (right column):

<instruction>
<examples>
Текст, который нужно проанализировать:
(К слову четвертое место в списке лидеров по итогам 2010 года занимает Москва).
Ответы экспертов к этому тексту:
[[’NULL’, ’Москва’, ’четвертое место в списке лидеров по итогам 2010 года занимает’, ’POS’]]
[]
[]
Ты можешь выбрать из этих ответов или ответить по-своему.
Твой ответ:

<instruction>
<examples>
Text to analyze:
(By the way, Moscow ranks fourth in the list of leaders based on the results of 2010.)
Expert answers to this text:
[[’NULL’, ’Moscow’, ’ranks fourth in the list of leaders based on the results of 2010’, ’POS’]]
[]
[]
You can choose from these answers or provide your own answer.
Your answer:

This stage can be repeated cyclically, i.e. the predictions of LLMs from this stage can be added to the prompt instead of predictions from the previous stage or in addition to them, and inference can be rerun with the new prompt.

To sum it up, the prompts have the following form:

Basic prompt (the first stage)	**Augemented prompt (the second stage)**
<instruction>	*<instruction>*
<examples>	*<examples>*
<target text>	*<target text>*
	<answers from the previous stage/iteration>

4 Experiments

4.1 LLMs

Originally we experimented with three super large language models: GPT-4o[1], Grok-Beta[2], and Mistral Large 2 [3]. As a baseline, we predicted opinion tuples with the basic prompt and randomly chosen examples from the train set (Stage 0 in Table 1).

[1]Model name via OpenAI API: `gpt-4o-2024-11-20`.
[2]Model name via xAI API: `grok-beta`.
[3]Model name via Mistral API: `mistral-large-2411`.

In the first stage we used the same prompt with examples chosen according to their semantic similarity to the target text. In the second stage the predictions of all three models were added to the basic prompt to form the augmented prompt. In the third stage we used the answers from the second stage to augment the prompt. Grok and GPT were tested only at stages 0–2.

We also experimented with DeepSeek-V3[4] but its answers were not included in the prompts (at stages 2–3), because we started experimenting with this model at a later moment. As will be shown below, this resulted in this model yielding the best score.

With each of the models, we set the temperature to 0.1, top-p to 0.9 and max tokens to 512. We used 12 examples per prompt. These parameters were chosen heuristically.

4.2 Text embeddings

We tried out two models for text embeddings generation: Sentence RuBERT[5] and SBERT[6]. We achieved slightly better results using Mistral with SBERT text embeddings during our preliminary experiments, so we used these text embeddings for example choice throughout the experiments we report here. However, we also experimented with switching to Sentence RuBERT embeddings after the second stage with two LLMs: Mistral Large 2 and DeepSeek-V3 (Stage 3b in Table 1). Intuitively, this makes sense because examples ranked as the most similar to the target by Sentence RuBERT and SBERT barely overlap. In particular, if 12 most similar examples are taken per target text, the average overlap across all 803 target texts is just 1.84 examples per text.

The results of our experiments are summarized in Table 1. The best result for each model is in bold. The overall best result is 0.363 achieved at Stage 2 by DeepSeek-V3. This was the second-best result in the competition by the time of submission of this paper (April 2024) with the best score being 0.41.

model	stage				
	Stage 0	Stage 1	Stage 2	Stage 3a	Stage 3b (w/ RuBERT)
gpt-4o-2024-11-20	0.239	**0.349**	0.344	–	–
grok-beta	0.245	0.33	**0.349**	–	–
mistral-large-2411	0.23	0.327	0.34	**0.345**	0.343
deepseek-chat	–	0.33	**0.363**	0.356	0.358

Table 1: F1-score on the test set

4.3 English prompt

Following a suggestion of an anonymous reviewer, we also partially replicated the experiments using the English version of the prompt, which can be found in Section 3 above. All else being equal, this resulted in a slight increase of F1-score in 3 out of 4 setups we had tried. This is shown in Table 2.

model	stage	
	Stage 1	Stage 2
mistral-large-2411	0.335	0.344
deepseek-chat	0.34	0.351

Table 2: F1-score on the test set using the English prompt

5 Discussion

Our overall findings are as follows:

1. The choice of semantically similar, rather than random, examples resulted in a significant increase of quality for each of the models.

[4]Model name via DeepSeek API: `deepseek-chat`.

[5]`http://hf.co/DeepPavlov/rubert-base-cased-sentence`

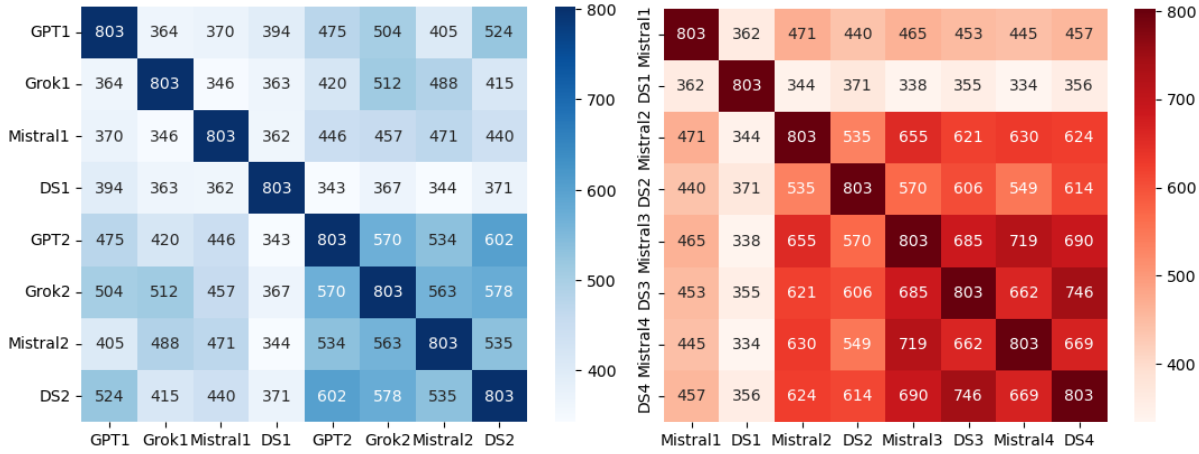
[6]`http://hf.co/ai-forever/sbert_large_nlu_ru`

2. The inclusion of answers from the previous stage improved the quality with 3 out of 4 models. The most significant increase is observed with DeepSeek-V3 which provided our best result in the competition. It should be noted once again, that we did not include the answers of this model in the augmented prompt, so this is the only model which was provided with the answers of three *other* models rather than its own answers and answers of two other models.
3. As for the third stage, it can be seen that it can improve the quality by only a very little margin, e.g. from 0.34 to 0.345 or 0.343 depending on text embeddings using Mistral. So, after the second stage further iterations of predictions using predictions from the previous stage does not seem to make much sense.
4. The best result was achieved with DeepSeek at the second rather than the third stage, although in both cases the model received completely novel information in the form of answer suggestions from Grok, GPT and Mistral and the suggestions at Stage 3 were of slightly better quality.
5. Finally, we observe that after the first stage the scores of three models which were prompted with each others answers converge at almost the same level (0.34–0.35 F1).

Figure 3 shows the matches between models' predictions. These heatmaps explain some of our observations. Firstly, as can be seen in Figure 3a, the models' predictions became more similar to each other in the second stage which explains similar scores achieved by different models at this stage (Observation 5). Secondly, it can be seen that the predictions of the first stage exhibit roughly the same level of disparity (upper left part of Figure 3a). However, DeepSeek's predictions in the first stage are very different from any model's prediction in the second stage. In other words, this LLM benefited from the augmented prompt more than the others and changed its predictions more radically which resulted in a higher improvement of the score. This is arguably due to the absence of DeepSeek's own predictions in the augmented prompt. This is a plausible explanation of Observation 2.

Figure 3a also explains Observation 4. Although predictions of Stage 2 are of higher quality, they are less variable than the predictions of Stage 1 in terms of content. Providing DeepSeek with more variable suggestions of slightly lower quality resulted in better predictions achieved at Stage 2.

As for Observation 3, Figure 3b shows that with further iterations of augmented prompting the models' predictions become increasingly similar which makes cyclic application of this procedure basically useless. In general, our results suggest that LLMs benefit from seeing answers generated by other LLMs but their own answers should not be mixed in.

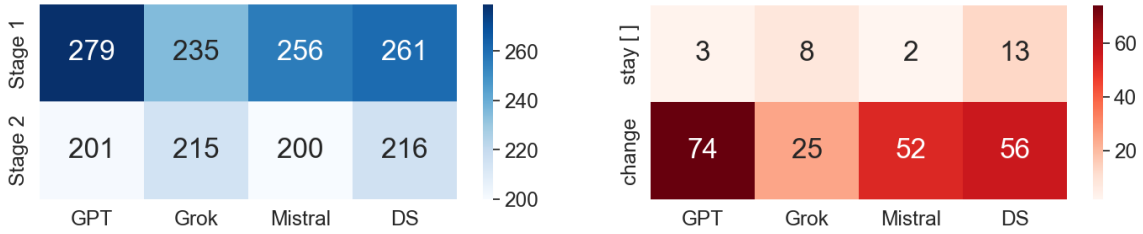


(a) All models, Stages 1–2

(b) Mistral & DeepSeek, Stages 1–3b

Figure 3: Pairwise counts of matching predictions of LLMs

We also observe that in the Stage 2 the number of predicted empty tuples dropped for all the models (Figure 4a). The models tend to switch to a non-empty tuple if at least one answer in the augmented prompt suggests a non-empty tuple. Figure 4b shows how the decisions were taken in such cases. This confirms that the answers suggested in the prompt have a significant influence on the models' output.



(a) Number of empty tuples predicted, Stages 1–2

(b) Resolution of conflicting predictions (Stage 2)

Figure 4: Changes in empty tuple predictions between Stage 1 and Stage 2

6 Conclusion

In this paper we described our participation in RuOpinionNE-2024 (Loukachevitch et al., 2025). Our method consisted in few-shot prompting of LLMs with semantically similar examples and answer suggestions. The contribution of this paper is two-fold. Firstly, we have shown that few-shot prompting of LLMs is a viable approach to SSA which allowed us to achieve the second-best score during the post-evaluation stage. Secondly, we explored interaction of super large language models in this task. In particular, we observed that (i) suggested answers have a significant influence on LLMs' predictions; (ii) LLMs benefit from novel and variable information from other LLMs; (iii) inclusion of LLMs' own answers and recycling of predictions in general do not have a positive impact.

Acknowledgments

The work was financially supported by Non-commercial Foundation for the Advancement of Science and Education INTELLECT.

We would like to thank the instructors at MSU.AI and in particular Alexander Ivchenko, Viktor Nemchenko and Artem Vasiliev. We would also like to thank Nikita Belyakov, Regina Nasyrova, Ksenia Studenikina, and anonymous reviewers of Dialogue 2025 for their valuable comments.

References

Jeremy Barnes, Robin Kurtz, Stephan Oepen, Lilja Øvrelid, and Erik Velldal. 2021. Structured sentiment analysis as dependency graph parsing. // *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processin*, P 3387–3402. Association for Computational Linguistics.

Jeremy Barnes, Laura Oberlaender, Enrica Troiano, Andrey Kutuzov, Jan Buchmann, Rodrigo Agerri, Lilja Øvrelid, and Erik Velldal. 2022. Semeval 2022 task 10: Structured sentiment analysis. // *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, P 1280–1295. Association for Computational Linguistics.

Anton Golubev, Nicolay Rusnachenko, and Natalia Loukachevitch. 2023. Rusentne-2023: Evaluating entity-oriented sentiment analysis on russian news texts. *Computational Linguistics and Intellectual Technologies*, 130–141.

Bing Liu. 2012. *Sentiment analysis and opinion mining*. Springer Nature.

Natalia Loukachevitch, Natalia Tkachenko, Anna Lapanitsyna, Mikhail Tikhomirov, and Nicolay Rusnachenko. 2025. Ruopinionne-2024: Extraction of opinion tuples from russian news texts. // *Proceedings of International Conference on Computational Linguistics and Intelligent Technologies Dialogue 2025*.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. Semeval-2013 task 2: Sentiment analysis in twitter. // *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, P 312––320. Association for Computational Linguistics, Atlanta, Georgia.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. // *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, P 79—86. Association for Computational Linguistics.

Maria Pontiki, Dimitros Galanis, John Pavlopoulos, Haris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. // *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014*, P 27–35, Dublin.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. // *Proceedings of the 2013 conference on empirical methods in natural language processing*, P 1631–1642, Seattle, Washington. Association for Computational Linguistics.

Peter D. Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. // *Proceedings of the 40th Annual Meeting of the Association for Computational Lin- guistics*, P 417–424, Philadelphia, Pennsylvania.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39:165–210.