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

April 23–25, 2025

An Approach to Information Extraction from Texts of a Limited Subject Domain Based on a Chain of Large Language Models

Sidorova E. A.
A.P. Ershov Institute of Informatics Systems; Siberian Branch of the Russian Academy of Sciences, Novosibirsk, Russia
lsidorova@iis.nsk.su

Ivanov A. I.
Novosibirsk State University, Novosibirsk, Russia
a.ivanov15@alumni.nsu.ru

Ilina D. V.
A.P. Ershov Institute of Informatics Systems; Siberian Branch of the Russian Academy of Sciences, Novosibirsk, Russia
dviljina@gmail.com

Ovchinnikova K. A.
Novosibirsk State University, Novosibirsk, Russia
k.ovchinnikova2@alumni.nsu.ru

Osmushkin N. M.
Novosibirsk State University, Novosibirsk, Russia
n.osmushkin@g.nsu.ru

Sery A. S.
A.P. Ershov Institute of Informatics Systems; Siberian Branch of the Russian Academy of Sciences, Novosibirsk, Russia
alexey.seryj@iis.nsk.su

Abstract

The paper considers the approach for extracting the information from texts of limited domains of knowledge based on a chain of neural language models. The task is represented in the form of three subtasks solved sequentially: (1) term extraction and classification; (2) coreference resolution; (3) extraction of relations of entities named with the terms. The dataset was based on texts on computational linguistics from the Habr forum. In the markup for term classification and relation extraction, 17 classes of terms and 51 relations were used in accordance with the ontology of computational linguistics. Prompt chain-based methods were used to apply LLMs, where each next query to the LLM is based on the results of the previous step. Six types of prompt templates were developed: for extracting, classifying, verifying terms, extracting coreferential relations, relations specified by the ontology, and a specialized template for relations linking entities of the same class. Sentence-BERT, GPT-4 and Mistral-based models were used at different steps of the study; a comparison with the SFT approach (ruRoBERTa) was made; hybrid approaches that have shown the best results were also developed. For term extraction and classification, F1=0.77 was obtained, for coreference resolution—F1=0.897, and for relation extraction—F1=0.847.

Keywords: information extraction; term extraction; relation extraction, natural language processing; machine learning; large language models; ontology of domain of knowledge; prompt chaining

DOI: 10.28995/2075-7182-2025-23-XX-XX

Подход к извлечению информации из текстов ограниченной предметной области на основе цепочки больших языковых моделей

Сидорова Е. А.
Институт систем информатики им. А.П. Ершова СО РАН, Новосибирск, Россия
lsidorova@iis.nsk.su

Иванов А. И.
Новосибирский государственный университет, Новосибирск, Россия
a.ivanov15@alumni.nsu.ru

Ильина Д. В.
Институт систем информатики им. А.П. Ершова СО РАН, Новосибирск, Россия
dviljina@gmail.com

Овчинникова К. А.
Новосибирский государственный университет, Новосибирск, Россия
k.ovchinnikova2@alumni.nsu.ru

Осьмушкин Н. М.
Новосибирский государственный университет, Новосибирск, Россия
n.osmushkin@g.nsu.ru

Серый А. С.
Институт систем информатики им. А.П. Ершова СО РАН, Новосибирск, Россия
alexey.seryj@iis.nsk.su

Аннотация

В статье рассматривается подход к извлечению информации из текстов ограниченной предметной области на основе цепочки нейросетевых языковых моделей. Задача представлена в виде трех подзадач, решаемых последовательно: (1) извлечение и классификация терминов; (2) разрешение кореференции среди терминов, найденных на первом этапе; (3) извлечение отношений между сущностями, обозначаемыми терминами. В качестве материала были взяты тексты по компьютерной лингвистике с форума Habr. Разметка текстов для классификации терминов и извлечения отношений проводилась в соответствии с типами сущностей и отношений, заданных онтологией. Всего при разметке датасетов использовалось 17 классов терминов и 51 отношение, относящихся к предметной области компьютерной лингвистики. Для применения больших генеративных языковых моделей (LLM) использовались методы на основе цепочки промптов, когда каждый следующий запрос к LLM генерируется на основе результатов предыдущего этапа. Было разработано 6 видов шаблонов промптов: для извлечения, классификации, верификации терминов, извлечения кореферентных отношений и отношений, заданных онтологией предметной области, а также специализированный шаблон для отношений, связывающий сущности одного класса. На разных этапах исследования применялись языковые модели Sentence-BERT, GPT-4 и модели на основе архитектуры Mistral; проведено сравнение с подходом на основе обучения (модель ruRoBERTa) и разработаны гибридные подходы, показавшие лучшие результаты. Достигнуты значения F_1=0.77 для извлечения и классификации терминов, F_1=0.897 для разрешения кореференции и F_1=0.847 для извлечения отношений.

Ключевые слова: извлечение информации; извлечение терминов; извлечение отношений; обработка естественного языка; машинное обучение; большие языковые модели; онтология предметной области; цепочки инструкций

1 Introduction

Information extraction (IE) focuses on extracting structured knowledge from natural language texts. Solutions are applied across various domains, including the analysis and monitoring of news, social networks, and scientific publications, as well as in automatic database population, and the construction of knowledge graphs and ontologies.

Ontologies and knowledge graphs are not only intended for human perception, but can also be used in information retrieval systems, monitoring, learning, etc. Subject domain ontologies play a special role when dealing with domain-specific texts, providing structures for representing information and, as a rule, offering the possibility to store the obtained information as a knowledge graph. They can also serve as a basis for generating prompts for LLMs in automatic text processing (NLP) systems.

In the paper, ontology is a formal representation of knowledge — whether abstract or specific — within a particular subject area. It is developed based on a description of objects, facts, and relationships and is designed for a versatile application across various tasks [1]. A knowledge graph refers to a semantic network that stores information about objects and the relationships between them [2].

Recently, Large Language Models (LLMs) have shown great promise in text understanding and generation. There has been a lot of work on the application of LLMs to IE tasks based both on discriminative and generative paradigms [3]. The main disadvantage of LLM of the first type is the need to prepare a representative dataset, which is not always possible in limited knowledge domains. A second-type LLM requires the selection and systematic refinement of instructions (prompts) in natural language, which also requires the involvement of experts.

Our study was motivated by the following factors:

- the models in few-shot and zero-shot settings still have a huge performance gap behind the SFT (Supervised Fine-Tuning);
- models perform significantly worse on limited subject areas (e.g., the GPT-4 performs 83.48 vs. 58.44 F-measure on the news corpus compared to the medical one [3]);
- despite the similarity of LLM-based architectures, the results vary greatly depending on the methods used, including design prompts;
- the results on the Russian-language dataset may differ significantly from those for the English-language dataset.

The aim of our study is to develop an LLM chain-based approach for solving the IE task from Russian-language texts with limited topics. For this, two research questions were formulated.

Q1. What level of quality can be achieved in IE from Russian-language texts within a subject area using zero-shot and few-shot methods based on generative LLM?

Q2. Can we combine the classic and LLM-based approaches for solving the IE task?

Experimental studies were carried out on a corpus of computational linguistics texts collected from the Habr forum (habr.com).

2 Related Work

The application of generative LLM for IE tasks can be called generative information extraction, since structural information is generated rather than extracted from text. To utilize an LLM, it is essential to formulate a query in natural language (a prompt), consisting of an instruction — a task for the model — and a text fragment — a context from which the model should extract information and generate the response.

Several approaches to generative IE can be distinguished.

- Decomposed prompting is a step-by-step, or specialized, approach when the solution is modeled sequentially using a chain of specialized models (Prompt Chaining) [4, 5]. The main subtasks are identified, and for each subtask a different prompt is developed, which can take into account the results obtained at the previous steps.
- A universal approach that assumes a unified prompt structure [6]. Chain-of-Thought Prompting (CoT) technique can be used to encode the information extraction scheme.
- QA prompt approach — a dialog approach in which an IE task is transformed into a multi-pass question-and-answer system [7].

The universal approach provides a simpler and faster solution by relying on a single query to a single LLM, avoiding the error accumulation associated with chain-of-solvers or dialog-based approaches. The dialog approach addresses the problem through multiple queries to a single LLM, implementing either a sequential chain or a reasoning tree. Both of them use prompts constructed from templates and previously extracted data, allowing for iterative corrections between the stages. The specialized approach assigns each task to a dedicated, fine-tuned model, enabling integration of various methods and the creation of hybrid solvers. This introduces a potential error-correction mechanism, enhancing the robustness and adaptability of the solution for complex reasoning tasks.

Prompts are task-specific instructions to guide the behavior of a model [8]. Advanced techniques of prompt engineering such as CoT, self-verification, self-consistency, meta-prompting, and generated

knowledge, significantly enhance model performance. In addition, it is noted that the use of several examples (few-shot prompting) also improves model performance compared to zero-shot or one-shot prompting.

According to recent studies, the performance of using LLMs for IE remains suboptimal [9]. Consequently, it is worth splitting IE to the multiple subtasks, considering them separately (Figure 1).

1. Named Entity Recognition (NER) involves extracting the boundaries of subject domain terms and classifying them according to the subject domain ontology. Terms are the basic concepts of the subject area necessary to describe its main phenomena, processes and events and constitute the terminological core of the ontology.

2. Coreference Resolution (CR) — establishing coreference relations between the terms found in the first step and clustering them to identify all mentions of each subject domain entity in a given text.

3. Relation Extraction (RE) — predicting semantic relationships between the entities found in the first two steps, and classifying them according to the relation types specified by the ontology.

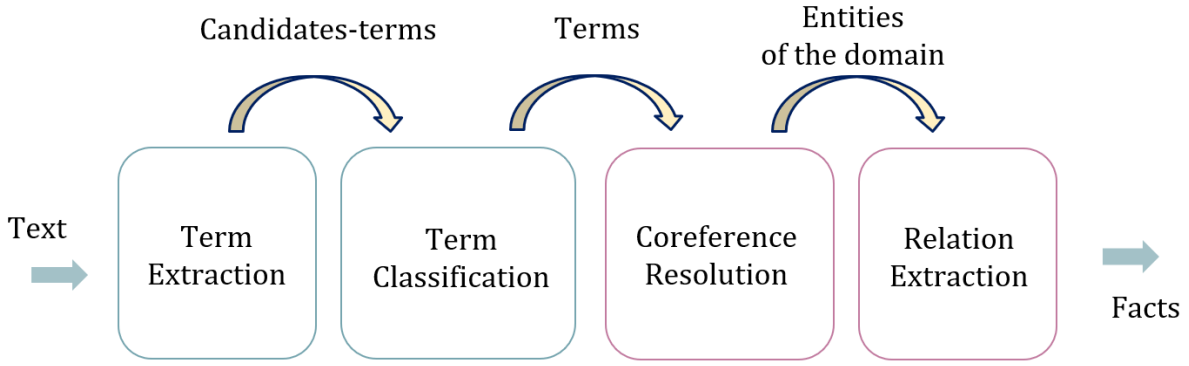


Figure 1: Main steps in information extraction

The application of LLMs to NER has been extensively studied, as this task is integral to many automatic text processing applications [10, 11]. However, solutions based on discriminative models and Supervised Fine-Tuning (SFT) still maintain an advantage over generative models, particularly in specialized knowledge domains. For instance, quality scores on the GENIA benchmark (biomedical data) reached 80–82% F_1-measure, whereas the best-performing generative model achieved 64.4% [12]. Furthermore, there is considerable variation in the quality of different generative models, highlighting the instability of these solutions.

Despite this, the generative-based approaches are still being actively explored [13]. In [14], two methods for generating prompts for the RE task are proposed: (a) based on knowledge of the possible types of relations between pairs of retrieved entities, and (b) using Question-Answering (QA) techniques. The quality of solutions to the relation classification task varies considerably, both depending on the dataset and across different models.

The CR task is also more efficiently addressed using SFT [15]. Although, typically additional entity information is not used, the authors of [16] prove that semantic information about entities enhances model predictions.

Competency questions (CQ) [17], which are essential to many ontology development methodologies, describe and constrain the domain of knowledge represented in the ontology. They define what the ontology should be able to answer. CQs can be generated automatically [18] and used as supplementary knowledge when composing prompts [19].

3 Preparing the data

Data preparation in IE tasks implies the creation of a representative corpus of texts of the subject domain, which serves as a basis for verification of results at all stages of the solving of the task. For this study, we collected a corpus of texts from the Russian-language website Habr related to the field of computational linguistics (CL) for the last 10 years, with a volume of about 1.5 thousand texts. The selection of necessary articles was based on the collected list of hubs. Based on this corpus, the subject area was analysed and datasets were created.

To capture the information important for a given subject domain (SD), formalise the structure of this information and all the data needed to retrieve it, a methodology was used to develop a subject domain ontology based on a base ontology whose concepts are specialised to more precise SD concepts and relations. In this paper, a top-level ontology on Computer Linguistics (CL) was used, which was obtained by instantiating the basic ontology of Scientific Knowledge [20]. The created ontology contains 17 classes and 111 relations. A fragment of this ontology is shown in Figure 2.

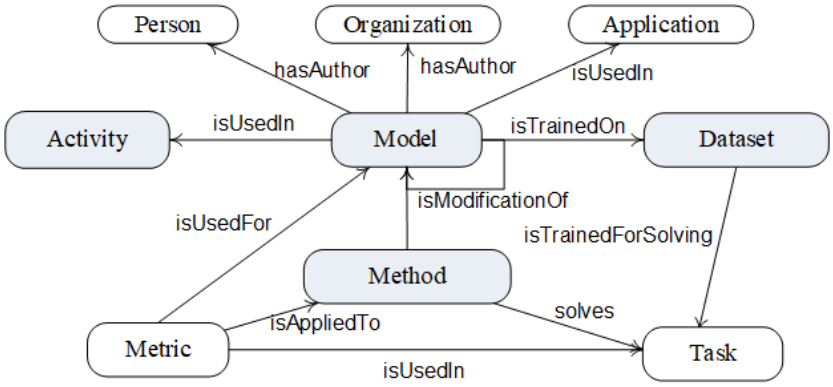


Figure 2: A fragment of the ontology on computational linguistics

To apply machine learning techniques to the IE tasks, we manually created a dataset by annotating terms, term classes, and the semantic relationships between terms. For the experimental studies, relations that were rarely mentioned in the texts were excluded from consideration. A total of 17 classes and 59 relations were used to mark up the datasets (the number of unique relation class names is 20; however, in the developed approach, the same-name relationships between different entity types are considered different, which allows for better analysis and evaluation of results).

To facilitate manual term annotation, the texts were initially processed using the SpaCy and a subject-specific dictionary. This dictionary was generated automatically from a top-level ontology, a thesaurus, and a web-portal on computational linguistics [20], and included 2640 words and word combinations related to the ontology classes.

The annotated relations were named based on the corresponding properties from the ontology, using the template <Subj.Property.Obj>, for example ***Model.isTraindOn.Dataset***. The format used in the CoNLL competition was used to annotate coreferential relations [15].

This resulted in the creation of two datasets.

a) The annotated corpus SentIE_CL contained 1088 informative sentences, in which 3136 terms and 1517 relations were annotated using the BIO format. The SentIE_CL dataset was used to train models using SFT.

b) Another dataset (DocIE_CL) is in json format and includes the markup of 4 full articles. The dataset containing 91 sentences, 606 terms, 232 semantic relations and 64 coreferences, was used for evaluation.

The created dataset and a complete list of the considered classes of terms and relations is presented at GitHub repository[1].

4 Approach to Information Extraction

To populate an ontology and build a knowledge graph, it is necessary to extract facts about entities of the subject domain defined in the ontology. The process of fact extraction included two steps: term extraction and relation extraction (Figure 1).

Based on the retrieved terms and their classes, entities (objects of a certain class) were created. Several terms may refer to the same entity, so before extracting the relations defined by the ontology, it was necessary to resolve coreference between the extracted terms (ideally also anaphora for pronouns). Next, the relations between terms were identified, and the attributes and relations (DataType Property and Object Property) between entities were established based on the relations type. It should be noted that

[1] https://github.com/Inscriptor/approach-to-IE-based-on-a-chain-of-LLMs.

for all the tasks considered, precision was prioritized, as the ultimate goal was to construct an accurate knowledge graph.

4.1 Comparative study of the approaches to Term Extraction

For the term extraction task, several LLM-based and prompt engineering approaches have been investigated and compared with the SFT-based approach, and two hybrid approaches have been considered. These approaches are an extension of the methods proposed by the authors earlier [20, 21].

The following instructions were developed for term extraction, classification and verification (Figure 3).

Term extraction

[INST]
Mark up the text using the BIO schema, using only the specified entity classes and their definitions.
{BIO-*description*}
Descriptions of available classes:
{*descriptions*}
Examples: ...
Use only the specified classes. Do not add new classes.
Examples of errors to avoid:
{errors}
Разметь текст по схеме BIO, используя только указанные классы сущностей и их определения. ...
Описания возможных классов: ...
Примеры: ...
Используй только указанные классы. Не добавляй новых классов.
Примеры ошибок, которых следует избегать: ...
[/INST]
Text: "{*text*}"
Response: <BIO-annotation>

Term classification

[INST]
You need to determine if the *{term}* is a term from *{domain}*.
Descriptions of available classes:
{*descriptions*}
Examples: ...
Don't give unnecessary explanations, answer briefly.
Тебе нужно определить, является ли словосочетание {*term*} термином из {*domain*}?
Descriptions of available classes: ...
Примеры: ...
Не давай лишних пояснений, ответь кратко.
[/INST]
Text: "{*text*}"
Word combination: {*term*}
Available classes: {*classes*}
Is "*{term}*" a term in this text?
Является ли "*{term}*" термином в этом тексте?
Response: <yes, class/no>

Term verification

[INST]
Your task is to determine whether an object is a member of *{class}* class or not.
Use the yes/no format.
{class} class description: *{description}*.
Give answers only within the text below.
Твоя задача состоит в том, чтобы определить, является ли объект представителем класса {*class*} или нет.
Используй формат да/нет.
Описание класса {*class*}: {*description*}.
Давай ответы только в рамках указанного ниже текста.
[/INST]
Text: {*text*}
Is *{term}* a term of the class *{class}* within this text?
Является ли {*term*} термином класса {*class*} в рамках этого текста?
Response: <yes/no>

Figure 3: Instruction templates for term extraction: translations are black, originals (in Russian) are grey

The ruRoBERTa model [22] was employed in the SFT-based approaches. Additionally, three Mistral-based models — Mistral (Mistral-7B-Instruct-v0.3), Nemo (Mistral-Nemo-Instruct-2407), and the Mixtral (Mixtral-8x7b-Instruct) [23] — along with GPT-4 [24], were utilized for prompt engineering. To obtain embeddings, the Sentence-BERT model (paraphrase-multilingual-mpnet-base-v2) [25] was applied.

The following approaches were considered.

1. **ruRoBERTa (identify) + ruRoBERTa (classify) —** training the roBERTa model on the SentIE CL data first to extract terms, then to classify them in a given context [20].
2. **GPT4 (extract)** and **Mixtral (extract)** — instruction-based term extraction.
3. **Mixtral (extract) + Mixtral (verify)** — extracting terms using instruction designed for extraction and then verifying using an instruction designed for verification.
4. **ruRoBERTa (identify) + ruRoBERTa (classify) + Mistral-based LLM (verify)** — applying the first model and verifying terms afterwards.
5. **LSP + Sentence-BERT + Vocabulary + Mistral-based LLM (classify) + Mistral-based LLM (verify)** — a hybrid method that integrates linguistic and neural network models; linguistic models are used to generate hypotheses and LLM-based approaches to test and refine them (Figure 4).

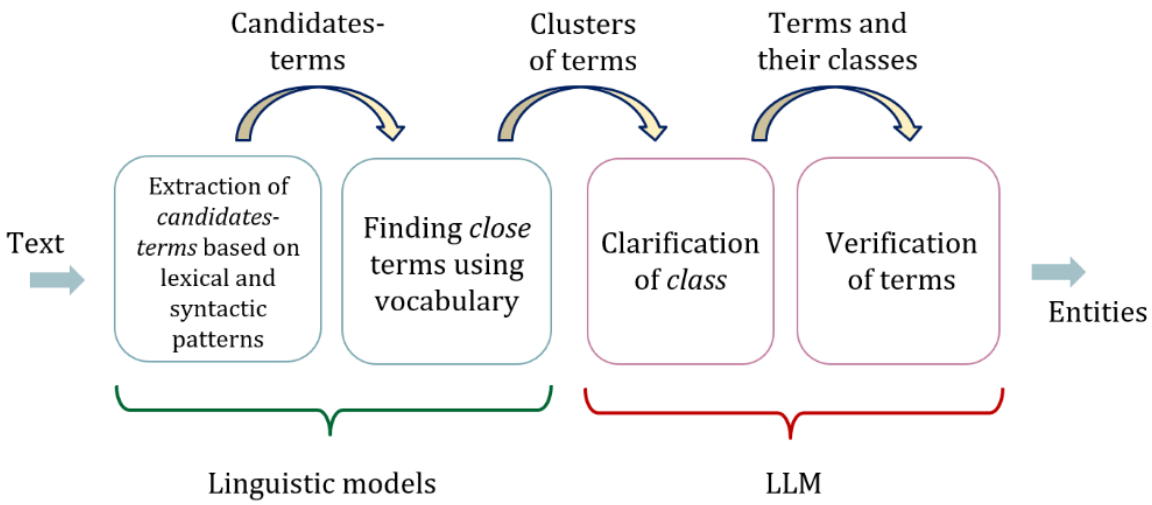


Figure 4: Hybrid approach to term extraction

The proposed hybrid approach for term extraction includes four components:

1. Lexico-syntactic patterns **(LSP)** allows the extraction of name groups based on 16 patterns.
2. **Vocabulary + Sentence-BERT** — subject dictionary and the method of comparison of embeddings obtained with the Sentence-BERT model allows to find for selected name groups close dictionary terms and on this basis to determine whether the new term belongs to the subject domain and to build hypotheses about its class (up to 6 classes).
3. **Mistral-based LLM (classify)** — predicting for each extracted term the most likely class from a set of predefined classes.
4. **Mistral-based LLM (verify)** — instruction-based verification of a term and its class.

Figure 5 shows examples of input and output data in the classification and verification stages of the suggested approach.

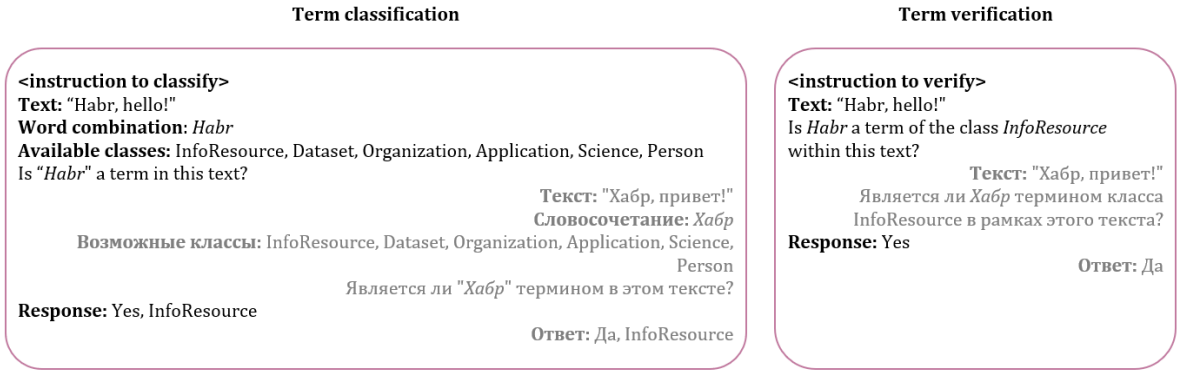


Figure 5: Example of input and output data in the classification and verification stages

In the example above, six available classes were proposed for the term *Habr* based on comparison with dictionary terms. At the first step, the model identified one class — ***InfoResource*** — using the prompt for classification. In the second step, the model performed a binary check of this result using the prompt for verification.

The obtained results for the above approaches are presented in Table 1.

Method	Recall	Precision	F_1
ruRoBERTa (identify) + ruRoBERTa (classify)	**0.81**	0.71	0.73
GPT4 (extract)	0.53	0.61	0.56
Mixtral (extract)	0.71	0.11	0.19
Mixtral (extract) + Mixtral (verify)	0.63	0.17	0.27
LSP + Vocabulary + Sentence-BERT + Mistral (classify) + Mistral (verify)	0.41	0.24	0.3
LSP + Vocabulary + Sentence-BERT + Nemo (classify) + Nemo (verify)	0.41	0.3	0.34
LSP + Vocabulary + Sentence-BERT + Mixtral (classify) + Mixtral (verify)	**0.78**	**0.71**	**0.74**
ruRoBERTa (identify) + ruRoBERTa (classify) + Mistral (verify)	0.44	0.35	0.39
ruRoBERTa (identify) + ruRoBERTa (classify) + Nemo (verify)	0.58	0.32	0.41
ruRoBERTa (identify) + ruRoBERTa (classify) + Mixtral (verify)	0.73	**0.89**	**0.77**

Table 1: Results of the term extraction and classifying

It can be concluded that adopting a comprehensive approach improved recall by leveraging templates and sufficiently large models, such as Mixtral. In contrast, the use of smaller, lightweight models has led to a decline in performance, primarily due to their lower classification quality. Using a combination of ruRoBERTa models for term extraction, ruRoBERTa for classification and LLMs for verification enabled the achievement of sufficiently high precision.

When analyzing the results, the following conclusions were made.

1. In the absence of verification, all methods extracted redundant terms, for which no corresponding ontology class existed.

2. A classifier may predict the term class differently depending on the context. In particular, some terms belonging to the ***Model*** class were incorrectly assigned to the ***Application*** class.

3. Challenges in identifying the complete phrase that is a term led to errors in extracting terms belonging to the ***Activity***, ***Dataset*** and ***Object*** classes. For instance, terms, such as *project model* instead of *project*, or *continuing experiments with optimizations* instead of *experiments* were assigned to the ***Activity*** category.

4.2 Coreference Resolution

Currently, the coreference resolution problem for the Russian language lacks robust solutions achieving performance comparable to the NER task [26], especially in restricted subject domains. As a result, significant limitations were introduced for the purposes of this study.

- Coreferential relations were sought only for the terms found (pronouns and name groups not related to subject domain terms were not considered).
- Relations were established only between terms of the same class (there was no deep hierarchy of classes in the subject domain).
- Coreference relations were established only within a single paragraph.

Coreference resolution was carried out using an LLM with an instruction designed for coreferential relation extraction. The model was provided with an instruction, the context in which a potential term-antecedent appeared, and additional auxiliary guidelines. Based on the model's responses, terms were clustered according to their coreferential relationships. Each resulting cluster corresponded to a distinct entity.

The results for some of the most representative classes of terms (considering only cases where the test dataset contained at least 30 occurrences of coreferential relations for each class) are in Figure 6. The mean F_1-measure for terms of all classes was 89.7%.

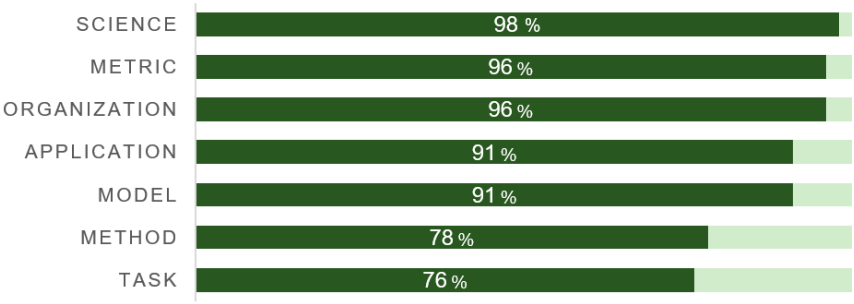


Figure 6: Coreference resolution results

4.3 Special Cases Study in Relation Extraction

The relation extraction involves identifying relations between the extracted entity terms. The set of possible relations is defined within the ontology, meaning that a relation between two terms is only possible if it exists between their respective classes in the ontology. Therefore, this task can be framed as a task of classifying the relation between a given pair of terms that are obtained in the previous steps and meet this requirement.

For each pair of terms, an instruction was generated, which included a text fragment (either a sentence or a paragraph) containing them, descriptions of the term classes, an explanation of the relation type to be tested, and examples of desired model responses (Figure 7).

Extraction of coreferential relations

[INST]
Your task is to determine whether or not the terms are names for the same entity in this text.
Answer only yes or no to the question below.
Твоя задача состоит в том, чтобы определять, являются термины названиями одной и той же сущности в этом тексте или нет.
Ответь только да или нет на вопрос ниже.
[/INST]
Text: {*text*}
Term {*class1*}: {*term1*}
Term {*class2*}: {*term2*}
Are {*term1*} and {*term2*} names of the same entity in this text?
Являются {*term1*} и {*term2*} названиями одной и той же сущности в этом тексте?
Response: <yes/no>

Relation extraction

[INST]
Your task is to determine whether a relationship exists between a pair of terms that belong to classes {*class1*} and {*class2*}.
The following questions may help you:
{*questions_list*}
Examples: {*examples_list*}
If a relationship exists, choose a relation for a pair of terms from the relation set {*relations_list*}. If the relation does not exist, write none.
Твоя задача определить, существует ли отношение между парой терминов, которые относятся к классам {*class1*} и {*class2*}.
Тебе могут помочь следующие вопросы: ...
Примеры: ...
Если отношение существует, выбери отношение для пары терминов из набора отношений {*relations_list*}. Если отношения нет, то напиши none.
[/INST]
Text: {*text*}
Term {*class1*}: {*term1*}
Term {*class2*}: {*term2*}
Is there an appropriate relationship between the terms {*term1*} and {*term2*} in this text?
Есть ли подходящее отношение между терминами {*term1*} и {*term2*} в этом тексте?
Response: <yes, relation/none>

Special relation extraction

[INST]
<...>
The terms should be arranged according to the meaning of the relation.
Термины должны быть упорядочены по смыслу отношения.
<...>
If a suitable relation exists, write the terms in the correct order so that it matches the direction of the relation from {*relation_names*}.
Если подходящее отношение есть, то запиши термины в правильном порядке, чтобы он соответствовал направлению отношения из {*relation_names*}.
[/INST]
Text: {*text*}
Term {*class1*}: {*term1*}
Term {*class2*}: {*term2*}
Is there an appropriate relation between the terms {*term1*} and {*term2*} in this text?
Есть ли подходящее отношение между терминами {*term1*} и {*term2*} в этом тексте?
Response: <yes, relation/no>

Figure 7: Instruction templates for relation extraction

Figure 8 shows the examples of the input and output data.

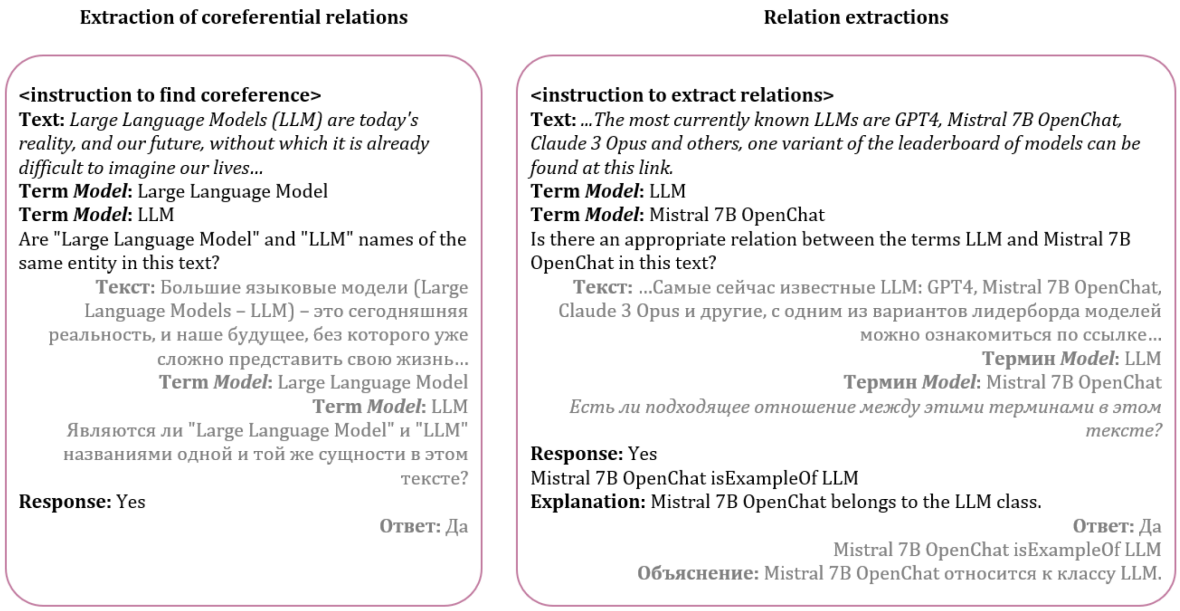


Figure 8: Examples of input and output data for the extraction of relations

In the example, the model receives as input the text fragment (sentence or paragraph) and two terms belonging to the class ***Model***. In the first case, the model calculates whether these terms are names of the same entity and gives a binary answer. In the second case, it answers the question whether there is a relationship between the given terms. The list of possible relations {relation_list} is computed when generating a prompt based on an ontology (a list of possible relations between given classes). In this case:

relation_list = {isModificationOf, isExampleOf, isPartOf, isAlternativeNameFor}

The model response includes three components: a binary response about the presence/absence of a relation, an indication of the type of relation, and an explanation of the response (this component was added later and performed better than the non-explanation response).

The average F1-measure was 84.7% (relations were extracted only within one sentence). The detailed results for all base types of relations are presented in [21].

A further refinement of the approach represented in the paper involves addressing cases that the model struggled to handle effectively. The primary complications arose from asymmetric relations between entities of the same class, such as inclusion relation (***Method.isPartOf.Method***) or example relations (***Method.isExampleOf.Method***). To improve the model's handling of these cases, an instruction for extracting asymmetric relations was developed, resulting in an increase in the precision. The change of recall and precision for the ***Model.isExampleOf.Model*** relation are detailed in Table 2.

Model.isExampleOf.Model	Recall	Precision
Before introducing the specialized instruction	0.50	0.28
After introducing the specialized instruction	0.21	**0.35**

Table 2: Results of relation extraction with standard and specialized instructions

The results presented in the table lead to a conclusion that while the precision increases significantly the recall decreases, which is in line with the expected, considering the ultimate goal of the ontology population, that was stated above. Therefore, the approach focused on improving the precision was considered justified and reasonable.

During the analysis, it was decided to subdivide the ***Model.isExampleOf.Model*** relation into two new types: hypo-hyperonymy and meronymy.

The ***Model.isExampleOf.Model*** relation is a hypo-hyperonymic relation that defines the relation between a model and the class of models to which it belongs.

The ***Model.isPartOf.Model*** relation is a meronymic relation that describes the part-to-whole relationship in situations where a model is part of another model in a literal sense, such as in the sentence, "The model created included a CRF layer" ("Созданная модель включала CRF-слой"). Once the relations were separated, their descriptions were also changed.

Splitting the ***Model.isExampleOf.Model*** relation also improved the extraction precision. The values of the metrics obtained after splitting the relation are shown in Table 3.

Relation	Recall	Precision	F_1
Model.isExampleOf.Model[2] *(before splitting into two relations)*	0.21	0.35	0.26
Model.isExampleOf.Model *(after splitting into two relations)*	**0.42**	**0.94**	**0.58**
Model.isPartOf.Model *(new relation)*	**0.33**	**0.60**	**0.43**
Model.isExampleOf.Model (Nemo) *(after splitting into two relations)*	0.12	0.34	0.17
Model.isPartOf.Model (Nemo) *(new relation)*	0.17	0.31	0.22

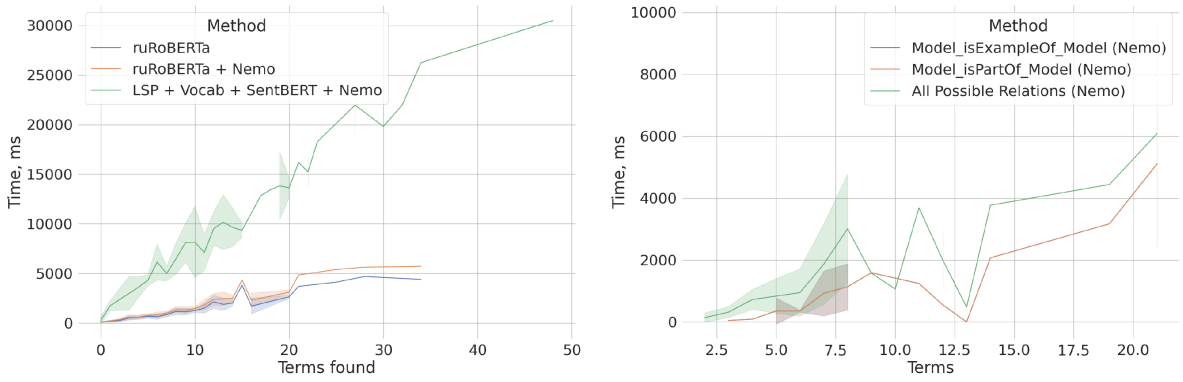
Table 3: Results of the term extraction and classifying

The values of the precision and recall metrics for the ***Model.isExampleOf.Model*** relation became higher. In addition, the values of the metrics for the ***Model.isPartOf.Model*** relation also seem to be quite high. Also, smaller models demonstrated significantly lower performance compared to the larger Mixtral model in the relation prediction task.

Figure 7 shows examples of real data sent with the prompts and received in response.

4.4 Evaluating the performance of the models

The approaches proposed by the authors are based on the application of transformer neural network models. Their computational complexity in comparison with recurrent and convolutional models is investigated in the original article [27].



a) Term Extraction b) Relation Extraction

Figure 9: Average text processing time

Figure 9 shows the average processing time per text for term extraction and the average time for predicting relations within a given set of terms. The average text volume is 383 characters or 52 words (the texts of the articles are divided into paragraphs).

The graphs are presented for local methods, i.e. methods that do not use external network resources, which showed the best results. The Mixtral model was called via the HuggingFace API, so most of the

[2] Unless otherwise specified, it is assumed that the Mixtral model was used.

time was spent on exchanging messages over the network. For this reason, the graphs for methods using this model are not shown in the figure. It is worth noting that the context length of the ruRoBERTa model is limited to 512 tokens, and instructions for large language models assume a simple response within a few tokens. This limits the potential number of terms that can be present in the text, and also simplifies the interaction with large language models, whose performance is highly dependent on the number of tokens that need to be generated in response to a query.

The term extraction time is shown as a function of the number of terms predicted by the initial model. This value affects both the text processing speed and the number of queries sent to the LLM. The graphs are presented for methods corresponding to methods **ruRoBERTa (identify) + ruRoBERTa (classify)**, **ruRoBERTa (identify) + ruRoBERTa (classify) + Nemo (verify)** и **LSP + Vocabulary + Sentence-BERT + Nemo (classify) + Nemo (verify)** in Table 1. As can be seen, the latter tends to predict a greater number of terms on average and, as a consequence, is a bit slower than methods based on the ruRoBERTa model.

The prediction time of relations on the example of relations ***Method.isExampleOf.Method***, ***Method.isPartOf.Method*** and all possible relations (see Table 2) is shown as a function of the number of terms of the corresponding classes in the given text, since a query to the LLM is formed for each pair of such terms. Coreference resolution was not considered an independent stage, since ***isAlternativeNameOf*** is considered a relation type. Therefore, the time required for coreference resolution is accordingly included in the total time for extracting all relations. As can be seen from Figure 9b, the plots for the relations ***Method.isExampleOf.Method***, ***Method.isPartOf.Method*** are almost identical.

Computations were performed on the following hardware configuration: AMD Ryzen 9 7950X, 32 GB RAM, NVidia GeForce RTX 4090 24 GB VRAM.

5 Conclusion

In this paper, an LLM-based information extraction approach (prompt chaining) was proposed and showed the following results: term extraction and classification (F_1=0.77, with the hybrid approach), simplified coreference resolution (F_1=0.89) and relation extraction (F_1=0.847). A new hybrid approach for term extraction integrating linguistic and neural network methods was proposed.

Although some of the problems were not solved, the research demonstrates that the use of LLM and few-shot prompting techniques can achieve results comparable to SFT methods. It is also noted that the involvement of an expert consistently improved the quality of the obtained solutions at every stage of the research: from the implementation of templates for assembling multi-word terms and the development of definitions for term classes to the development of instructions for extracting special relations.

The development of the approach can consist of applying data fusion techniques that show consistently good results on the NER task [28], as well as applying various techniques for automatic prompt improvement.

References

[1] Loukachevitch Natalia V., Dobrov Boris V. (2015), Developing Linguistic Ontologies in Broad Domains [Proektirovanie lingvisticheskikh ontologiy dlya informatsionnykh sistem v shirokikh predmetnykh oblastyakh], Ontology of Designing [Ontologiya Proektirovaniya], Vol. 5, no. 1(15), pp. 47–69.

[2] Ehrlinger L, Wöß W. Towards a Definition of Knowledge Graphs. Joint Proc. of the Posters and Demos Track of 12th Int. Conf. on Semantic Systems (SEMANTiCS2016) and 1st Int. Workshop on Semantic Change & Evolving Semantics (SuCCESS16). — Leipzig, Germany. — 2016. — P. 13–16.

[3] Derong Xu, Wei Chen et al. Large Language Models for Generative Information Extraction: A Survey // Frontiers of Computer Science. — 2024. — Vol. 18, no. 6. — Article no. 186357. — https://doi.org/10.1007/s11704-024-40555-y.

[4] Shichao Sun, Ruifeng Yuan, Ziqiang Cao, Wenjie Li, Pengfei Liu. Prompt Chaining or Stepwise Prompt? Refinement in Text Summarization // Findings of the Association for Computational Linguistics ACL 2024. — Bangkok, Thailand and virtual meeting. — 2024. — P. 7551–7558. — https://doi.org/10.18653/v1/2024.findings-acl.449.

[5] Khot Tushar, Trivedi Harsh et al. Decomposed Prompting: a Modular Approach for Solving Complex Tasks // Computing Research Repository. — 2022. — Vol. ArXiv:2210.02406. — version 2. Access mode: https://arxiv.org/abs/2210.02406.

[6] Yaojie Lu, Qing Liu et al. Unified Structure Generation for Universal Information Extraction. // Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). — Dublin, Ireland. — 2022. — P. 5755–5772. — https://doi.org/10.18653/v1/2022.acl-long.395.

[7] Xiang Wei, Xingyu Cui et al. Zero-Shot Information Extraction via Chatting with ChatGPT // Computing Research Repository. — 2023. — Vol. arXiv:2302.10205. — version 2. Access mode: https://arxiv.org/abs/2302.10205.
[8] Pengfei Liu, Weizhe Yuan et al. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing // ACM Computing Surveys. — 2023. — Vol. 55, iss. 9. — P. 1–35. — https://doi.org/10.1145/3560815.
[9] Ridong Han, Tao Peng et al. Is Information Extraction Solved by ChatGPT? An Analysis of Performance, Evaluation Criteria, Robustness and Errors // Computing Research Repository. — 2023. — Vol. arXiv:2305.14450. — version 2. Access mode: https://arxiv.org/abs/2305.14450v2.
[10] Ashok D, Lipton Z C. PromptNER: Prompting for Named Entity Recognition // Computing Research Repository. — 2023. — Vol. arXiv:2305.15444. — ersion 2. Access mode: https://arxiv.org/abs/2305.15444.
[11] Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, Hongwei Wang. Empirical Study of Zero-Shot NER with ChatGPT // Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Singapore. — 2023. — P. 7935–7956. — https://doi.org/10.18653/v1/2023.emnlp-main.493.
[12] Wang S, Sun X, Li X, Ouyang R, Wu F, Zhang T, Li J, Wang G. GPT-NER: Named Entity Recognition via Large Language Models // Computing Research Repository. — 2023. — Vol. arXiv:2304.10428. — version 4. Access mode: https://arxiv.org/abs/2304.10428.
[13] Zhen Wan, Fei Cheng et al. GPT-RE: In-context Learning for Relation Extraction using Large Language Models // Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. — Singapore. — 2023. — P. 3534–3547. — https://doi.org/10.18653/v1/2023.emnlp-main.214.
[14] Kai Zhang, Gutierrez Bernal Jimenez, Yu Su. Aligning Instruction Tasks Unlocks Large Language Models as Zero-Shot Relation Extractors // Findings of the Association for Computational Linguistics: ACL 2023. — Toronto, Canada. — 2023. — P. 794–812. — https://doi.org/10.18653/v1/2023.findings-acl.50.
[15] Coreference Resolution on CoNLL 2012 // Paper with Code Repository. — Access mode: https://paperswithcode.com/sota/coreference-resolution-on-conll-2012.
[16] Khosla Sopan, Rose Carolyn. Using Type Information to Improve Entity Coreference Resolution // Proceedings of the First Workshop on Computational Approaches to Discourse. November 20, 2020. — Online. — 2020. — P. 20–31. — https://doi.org/10.18653/v1/2020.codi-1.3.
[17] Wiśniewski Dawid, Potoniec Jedrzej, Ławrynowicz Agnieszka, Keet Catharina Maria. Analysis of Ontology Competency Questions and their formalizations in SPARQL-OWL // Journal of Web Semantics. — 2019. — Vol. 29. — P. 105098. — https://doi.org/10.1016/j.websem.2019.100534.
[18] Pan Xueli, van Ossenbruggen Jacco, de Boer Victor, Zhisheng Huang. A RAG Approach for Generating Competency Questions in Ontology Engineering // Computing Research Repository. — 2024. — Vol. arXiv:2409.08820. — version 1. Access mode: https://arxiv.org/abs/2409.08820.
[19] Jiacheng Liu, Alisa Liu et al. Generated Knowledge Prompting for Commonsense Reasoning // Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). — Dublin, Ireland. — 2022. — P. 3154–3169. — https://doi.org/10.18653/v1/2022.acl-long.225.
[20] Ovchinnikova K., Ivanov A., Sidorova E. Automation of the construction of the terminological core of ontology in computer linguistics based on a corpus of texts [In Russian] // System Informatics. — 2023. — No. 23. — P. 13–32.
[21] Sidorova E., Ivanov A., Ovchinnikova K. Information extraction from texts based on ontology and large language models [In Russian] // Ontology of designing. 2025. — Vol. 15, no. 1. — P. 114–129. — https://doi.org/10.18287/2223-9537-2025-15-1-114-129.
[22] Zmitrovich D., Abramov A., Kalmykov A., Tikhonova M., Taktasheva E., Astafurov D., Baushenko M., Snegirev A., Shavrina T., Markov S., Mikhailov V., Fenogenova A. A Family of Pretrained Transformer Language Models for Russian // Computing Research Repository. — 2023. — Vol. arXiv:2309.10931. — version 4. Access mode: https://arxiv.org/abs/2309.10931.
[23] Mistral AI // Hugging Face Repository. — Access mode: https://huggingface.co/mistrala.
[24] GPT4-Turbo and GPT-4 // OpenAI. — Access mode: https://platform.openai.com/docs/models#gpt-4-turbo-and-gpt-4.
[25] paraphrase-multilingual-mpnet-base-v2 // Hugging Face Repository. — 2021. — Access mode: https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2.
[26] Dobrovolskii Vladimir, Michurina Mariia, Ivoylova Alexandra. RuCoCo: a new Russian corpus with coreference annotation // Computing Research Repository. — 2022. — Vol. arXiv:2206.04925. — version 1. Access mode: https://arxiv.org/abs/2206.04925.
[27] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł., Polosukhin I. Attention is all you need // Advances in neural information processing systems. — 2017. — No. 30. — P. 5998–6008. — https://doi.org/10.48550/arXiv.1706.03762.
[28] Hu X, Jiang Y, Liu A, Huang Z, Xie P, Huang F, Wen L, Philip S Y. Entity-to-Text based Data Augmentation for various Named Entity Recognition Tasks // Findings of the Association for Computational Linguistics: ACL 2023. — Toronto, Canada. — 2023. — P. 9072–9087. — https://doi.org/10.18653/v1/2023.findings-acl.578.