big data and cognitive computing

MDPI

Article

Transferring Natural Language Datasets Between Languages Using Large Language Models for Modern Decision Support and Sci-Tech Analytical Systems

Dmitrii Popov [1,2,3], Egor Terentev [1,2], Danil Serenko [1,2], Ilya Sochenkov [1,3,4] and Igor Buyanov [1,*]

1 Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences (FRC CSC RAS), Moscow 119333, Russia; popov.dmitriy.p@yandex.ru (D.P.); eterentevd@yandex.ru (E.T.); serenko.d.s@yandex.ru (D.S.); sochenkov@isa.ru (I.S.)
2 Faculty of Physics and Mathematics and Natural Sciences, RUDN University, Moscow 117198, Russia
3 Institute for Information Transmission Problems of the Russian Academy of Sciences (IITP RAS), Moscow 127051, Russia
4 Ivannikov Institute for System Programming of the Russian Academy of Sciences (ISP RAS), Moscow 109004, Russia
* Correspondence: buyanov.igor.o@yandex.ru

Abstract: The decision-making process to rule R&D relies on information related to current trends in particular research areas. In this work, we investigated how one can use large language models (LLMs) to transfer the dataset and its annotation from one language to another. This is crucial since sharing knowledge between different languages could boost certain underresourced directions in the target language, saving lots of effort in data annotation or quick prototyping. We experiment with English and Russian pairs, translating the DEFT (Definition Extraction from Texts) corpus. This corpus contains three layers of annotation dedicated to term-definition pair mining, which is a rare annotation type for Russian. The presence of such a dataset is beneficial for the natural language processing methods of trend analysis in science since the terms and definitions are the basic blocks of any scientific field. We provide a pipeline for the annotation transfer using LLMs. In the end, we train the BERT-based models on the translated dataset to establish a baseline.

Keywords: large language model; machine translation; data transferring; ChatGPT; Llama; DeepSeek; Qwen; DEFT

check for updates

Academic Editor: Domenico Ursino

Received: 24 March 2025
Revised: 18 April 2025
Accepted: 23 April 2025
Published: 28 April 2025

Citation: Popov, D.; Terentev, E.; Serenko, D.; Sochenkov, I.; Buyanov, I. Transferring Natural Language Datasets Between Languages Using Large Language Models for Modern Decision Support and Sci-Tech Analytical Systems. *Big Data Cogn. Comput.* **2025**, *9*, 116. https://doi.org/10.3390/bdcc9050116

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In this paper, we develop our approach by extracting definitions of terms being used or introduced. Extracting definitions helps to track the continuity and frequency dynamics of terminology used in scientific and technical documents over time. In the absence of large-scale Russian corpora with labeled terms, the problem is relevant. The results of this paper can be used as the basis for terminology extraction tools in analytical systems such as iFORA (https://issek.hse.ru/en/ifora/, accessed on 22 April 2025), SciApp (https://sciapp.ru/, accessed on 22 April 2025), Neopoisk (https://neopoisk.ru/publ/, accessed on 22 April 2025), and others. Information plays a crucial role in the decision-making process to rule R&D in research institutions, universities, and companies. As the information overhead grows year by year, the natural language processing community is challenged to bring a method that allows orientation in this endless amount of papers. To stay on the cutting edge of the field or quickly get to know the new one, it is nice to have some ways to make it easier. One can say that terms and their definitions are the basic blocks of any

study. It would be a handful to have a method that can extract them from literature in order for the learner to get familiar with them. It is also useful when analyzing a large amount of scientific data. To analyze the development of scientific and technological directions, it is important to identify the terminology used and introduced by the authors. Classical approaches in scientific and technical analytics use methods of extracting terminology from texts [1–3] in the form of words and phrases.

The mainstream approach for classification problems is to use neural networks. Their ability to model complex, nonlinear relationships in data makes them highly effective wide range of applications, including terminology and its definition extraction. Their architecture allows them to learn hierarchical feature representations from raw input, improving performance with increased data and computational resources. Additionally, advances in deep learning, such as attention mechanisms and Transformer architecture, have significantly enhanced their capability to handle text data. The availability of large datasets and improvements in hardware, particularly GPUs, have further facilitated the training of deep neural networks, making them a preferred choice. Lastly, extensive open-source libraries and community support have accelerated their adoption and implementation across various fields. However, they require a large amount of annotated data to be trained. The vast number of parameters in neural networks, especially deep learning models, necessitates extensive training data to prevent overfitting and to generalize well to new, unseen data. Labeled data provides the ground truth that helps the network adjust its weights effectively during training, leading to improved accuracy and performance. Additionally, the diversity present in large datasets aids in capturing the variability in real-world scenarios, ensuring the model's robustness. Insufficient labeled data can lead to poor model performance, making extensive datasets crucial for successful neural network training.

One of the recent resources dedicated to the abovementioned task is the DEFT corpus [4]. Developing for the SemEval 2020 task [5] consists of the English texts from free e-books with tree-layer annotation: whether the text has a definition, annotation of terms and definitions as named entities, and relations between them. The problem is that other languages lack such resources, such as Russian. It would be great to somehow automatically transfer the existing English DEFT dataset into other languages to obtain a starting point. Further, such a transferred dataset could be corrected by human annotators, which is easier and cheaper than crafting the dataset from scratch.

While general text classification annotation could be transferred by the language translation, the transferring of the named entity annotation from one language to another is challenging since the matched spans in the source and target languages must be found. Recently, the large language models (LLMs) have demonstrated high effectiveness in a broad variety of NLP tasks. Unlike traditional NLP methods that rely heavily on manual feature engineering and rule-based systems, LLMs leverage the Transformer architecture to learn patterns and contextual relationships from vast amounts of text data. This approach allows LLMs to perform tasks such as translation [6] and named entity recognition (NER) [7] with greater efficiency and adaptability across different languages and contexts. LLMs surpass traditional methods by eliminating the need for extensive task-specific programming and by excelling in zero-shot and few-shot learning scenarios, where little to no task-specific data are available. They bring improvements in scalability, as they can be fine-tuned for numerous applications with minimal adjustments, and they enhance performance by capturing nuanced language subtleties that traditional models may overlook. Furthermore, LLMs have shown the ability to generalize across tasks, providing a unified model capable of addressing diverse NLP challenges. This versatility reduces the need for multiple specialized systems, ultimately streamlining the development process. Motivated by these advantages, we utilize LLMs for the cross-language annotation transfer by using them as a

"smart" translator that can preserve the named entity spans while translating the text. We hypothesize that they can automate the process of adaptation of an annotated dataset from different languages, allowing one to obtain a quick baseline or giving a solid start in the annotation process. We limit this work only to English–Russian language pairs, leaving other languages for future work.

To summarize, the contribution of our work is as follows:

- We show how we transfer the NER annotation using LLMs on the English and Russian language pairs.
- We analyze the translation quality of several modern LLMs from English to Russian for this particular task. It includes ChatGPT, Llama3.1-8B [8], DeepSeek, and Qwen.
- We provide the result of the BERT-like models trained to make a baseline for two of the three original DEFT tasks: detection of texts with definitions and named entity recognition. In addition, we provide the results of our pipeline for the Wikipedia part of the WCL dataset.

We opensourced the datasets (https://huggingface.co/datasets/astromis/ruDEFT, accessed on 22 April 2025, https://huggingface.co/datasets/astromis/WCL_Wiki_Ru, accessed on 22 April 2025) and code (https://github.com/Astromis/research/tree/master/rudeft, accessed on 22 April 2025).

2. Related Work

The task of term and definition extraction has a long story because it is tightly related to the desire to structure the information from various texts. Starting from rule-based systems [9], which relies on handcrafted rules, it evolves to statistical methods [10–12] and lastly to a deep learning system [13,14]. The statistical methods operate either by automatically mining the patterns from the dataset or by constructing a set of features that fit into a machine learning model. The deep learning methods fully rely on neural networks like LSTM [15] or, recently, Transformer-based architecture [16].

Frequently, statistical or deep learning-based papers come with their datasets. The WCL dataset [17] was developed as a part of the work in [10]. This is a dataset with annotated definitions and hypernyms composed of Wikipedia pages and a subset of the ukWaC Web corpus. The SymDef dataset comes from [18] and approaches the problem of bounding the mathematical symbols with their definitions in scientific papers parsed from arXiv. Speaking about the lack of resources, it is worth mentioning that for the Russian language, it is easy to find only the RuSERRC dataset [19] in which the terms were annotated. The Russian datasets with definition annotation are unknown.

With the rise of the LLMs, researchers have started to investigate the method of using them in dataset annotation or generation. In the work [20] researchers employ ChatGPT to be an annotator with an explain-than-annotate technique. They compared its performance with crowdsourcing annotation and obtained promising results that ChatGPT is on par with crowdsourcing. Regularly, researchers issue the best practices about how to obtain the best from LLMs as annotators, like in [21]. LLMs are used in interesting annotation projects, like creativity dataset creation [22], where the LLM produces some ideas of how to use an unrelated set of items to solve a particular task, and humans only verify these ideas. In some cases, the LLM is used directly for dataset synthesis [23,24].

3. Materials and Methods

3.1. DEFT Corpus

The DEFT corpus is a collection of text from free books available on https://cnx.org/ (accessed on 22 April 2025). The texts cover topics like biology, history, physics, psychology, economics, sociology, and government. NER annotation includes the term and definition

labels as primary annotation and supportive annotation for the cases as aliases, orders, and referents for both terms and definitions. For the task of detecting a sentence that contains a definition, the annotation is produced straightforwardly from the presence of the definition NER annotation in a sentence. The relation annotation bounds terms and definitions. For a detailed description of the tags, annotation process, and challenges, we refer to the original paper. We use the available DEFT corpus on GitHub (https://github.com/adobe-research/deft_corpus, accessed on 22 April 2025, version of the corpus from 16 January 2020, commit is db8c95565c2e58d861537cb8cb4621c50b75cd13). The entity statistics are provided in Figure 1 in "ENG" legend parts for train, dev test, and the prepared gold set, which we talk about in the section "Preparing the gold set". We also want to point out that we will not experiment with a third annotation with a relation between terms and definitions. We leave it for future work.

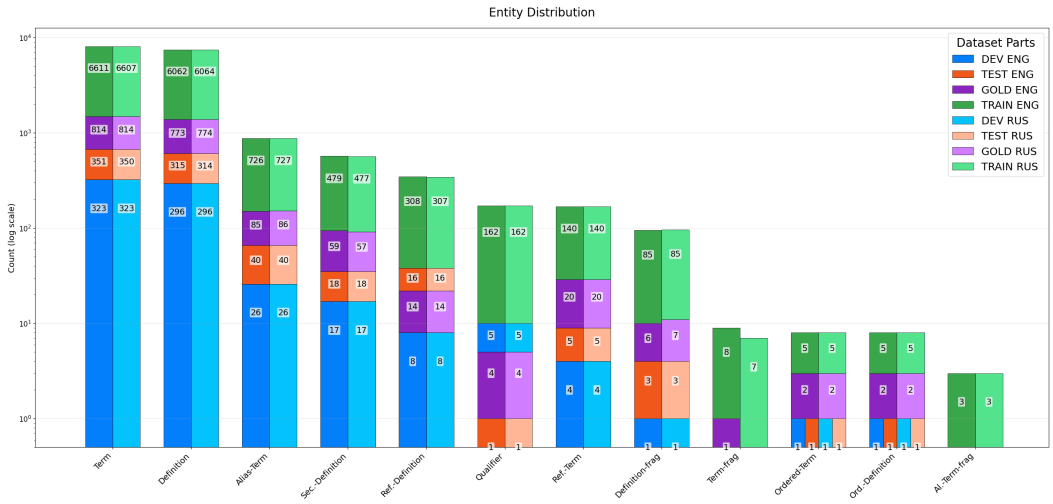

Figure 1. The statistics of the entities in the DEFT corpus. The gold part shows the part prepared by us for testing the LLMs.

To work with our pipeline, we had to convert the original CoNLL-like (Conference on Natural Language Learning) format into the Hugging Face Datasets library [25]. Basically, the dataset can be represented as a list of dictionaries (objects) that can be easily converted to and from JSON. In turn, it makes it easy to communicate with LLMs in the latter format.

We noticed two issues while preprocessing the original files of the corpus. The first one is a tokenization error when two sentences containing the wide span are wrongly separated. As a consequence, the second sentence starts with the token having an "I" tag, which is illegal in the IOB (inside, outside, beginning) format.

While the first issue was found just once, the second issue with data duplicates occurs more often. We count 2187 duplicate sentences. Moreover, these duplicates have different annotations. It seems that the merging error occurred when the corpus was being compiled.

3.2. WCL Corpus

In addition, to make our research broader, we apply our pipeline to the Wikipedia part of the WCL dataset. We chose this part because of a clear understanding of the structure, where all data were divided into two files: one file contains sentences with definitions and another file contains just regular sentences. All these sentences have an annotation of the term token, but not for the definition itself.

We convert the dataset from its original format to the common Hugging Face structure, resampling what we obtain from converting the DEFT dataset. All in all, we obtain 2822 sentences with no definitions and 1869 sentences with definitions. We next divide the whole dataset into train, dev, and test splits in the proportion 70/10/20.

Nevertheless, our main focus is the DEFT dataset.

3.3. Large Language Models

We benchmarked a diverse set of state-of-the-art LLMs to assess their performance on our tasks. The specific model checkpoints evaluated are as follows:

- llama-3.1-8b-instruct
- gpt-3.5-turbo
- gpt-4o-mini
- gpt-4.1-nano
- gpt-4.1-mini
- deepseek-chat-v3-0324
- deepseek-r1
- qwen-2.5-72b-instruct

To simplify the experimental setup and ensure reproducibility, we leveraged the bothub.chat service (https://bothub.chat/, accessed on 22 April 2025) as a unified proxy for accessing the aforementioned models. This service provides a streamlined interface to various APIs—including OpenAI's ChatGPT, DeepSeek, Llama, and QWEN—thereby abstracting the need for direct API integration. This approach not only facilitated rapid testing and experimentation but also allowed for systematic documentation of any associated computational costs, which were primarily linked to underlying API usage fees. Detailed statistics on time and monetary expenditures are provided in Appendix E.

3.4. Methodology

On a high level, the methodology consists of three steps: preparing the gold set, automatic translation, and annotation transfer. We describe each of them in separate paragraphs. The overview of the methodology steps is visualized in Figure 2.

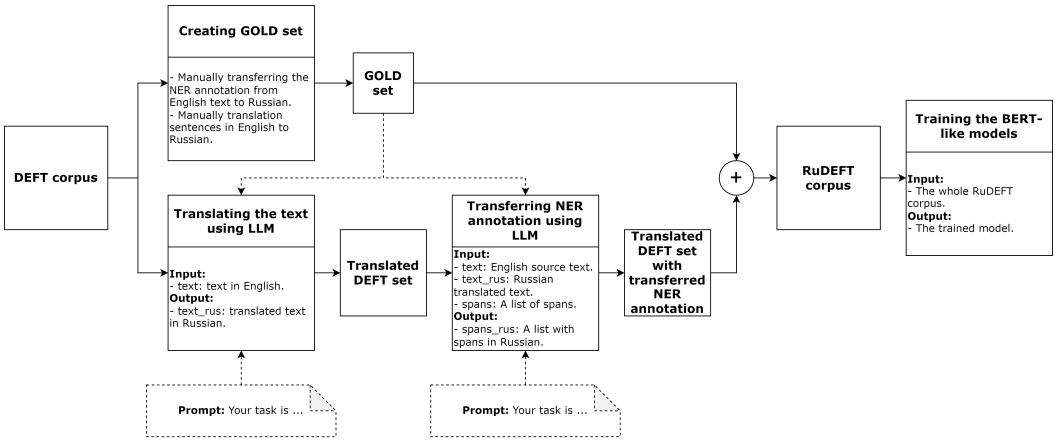

Figure 2. The methodology steps.

3.4.1. Preparing the Gold Set

To be able to estimate the output quality of our steps, we need a reliable set that was manually checked in terms of the translation and NER annotation. To do that, we translate the whole dev set and a small part of the train set with the API Google Translate. We obtain 1179 sentences from the dev set and 3010 out of 24,184 randomly selected sentences from the train set. (There is no specific reason why exactly 3010 from the train part were sampled; it just happened once, and we decided to let it be). Next, we select only the sentences with NER annotation, which gives us 870 sentences from the dev and train sets. Then we manually transfer the NER annotation from English text to Russian using Label Studio [26]. While transferring, when we saw that the translation was semantically

incorrect, we skipped these sentences and later translated them manually. The source of incorrectness originates commonly from the catchphrases and the specific language. The statistics of the NER labels of the gold set are available in Figure 1.

For the purpose of evaluating the translation, we sampled 200 sentences with no NER annotation and also checked them for adequate translation.

Finally, the statistics of our gold set are next: 870 sentences have the NER annotation, and 200 sentences do not. Overall, we have 1070 sentences with manually verified translations into Russian.

3.4.2. Transferring NER Annotation Using LLMs

For the task of NER annotation transferring, we test several LLMs, such as Llama3.1-8B and Qwen-2.5-72B, variants of DeepSeek, and variants of GPT-4 and ChatGPT3.5-turbo from OpenAI. They vary in scale, which directly influences the cost and generation time. The latter is crucial when one works with a large amount of data. Also, while OpenAI's models are closed-sourced, it is of high interest how the open-sourced models such as Llama3.1, Qwen, or DeepSeek are suitable for such tasks, as many researchers and companies cannot rely on third-party API because of data privacy.

As a note, we use Qwen2.5-72B because it was released during this work, so we decided to include a bigger LLM of the fresh release and not Llama3.1-72B.

To build the prompt, we try several standard prompt techniques like zero-shot, few-shot, and chain-of-thoughts [27]. We select 20 examples from the gold dataset in the early stages of the prompt development. That means we reject some prompt-building strategies if they cannot deal with most of this subset. The only prompts that achieve more than 15 cases of correct annotation transferring will be selected for further testing.

At first, we formulate the task for the LLM as follows: given the source English text and the list of annotation span triplets consisting of start index, end index, and label name, we ask the model to find in a given Russian text span triplets that correspond to English ones. Unfortunately, this approach failed to give a good output quality, as we show in the "Results" section. So, we reformulated the task to find a substring in Russian that corresponds to a substring in English that is actually an NER span. See Figure 3 for a visual explanation of the resulting approach.

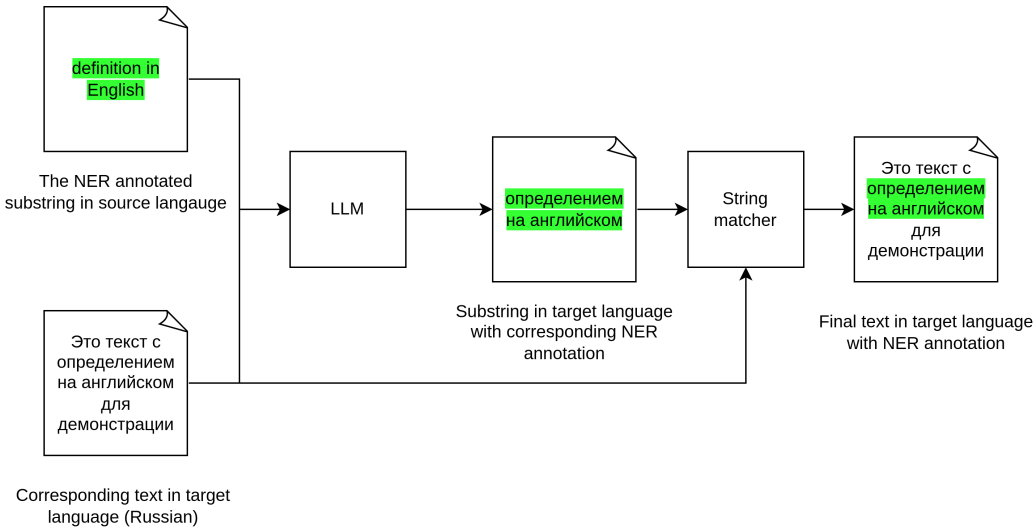

Figure 3. The step-by-step illustration of the NER annotation transferring. The green highlight shows the NER annotation.

After obtaining a prompt that beats the simple test, we evaluate it on the whole gold dataset, where we manually transfer the annotation. As a metric, we use the number of matches between the gold transfer and LLM in different situations:

- Exact matches count the cases when the indices of the gold and transferred spans are equal;
- Wider partial matches count the cases when the transferred span is wider than the original one;
- Narrower partial matches count the cases when the transferred span is narrower than the original one;
- Mismatches are self-explained cases;
- Total spans checked accounts for the processed cases. Note that it differs between LLMs, as some examples could not be handled correctly, even after several retries.

3.4.3. Translating the Text Using LLMs

The translation task is pretty straightforward. Given the text in English, we ask the LLM to translate it into Russian. From the previous research [6], we know that LLMs are good enough in this task, though they still perform worse than supervised systems. Nevertheless, we are interested in building a monolithic pipeline based solely on LLMs. To ensure the quality of the translation, based on our gold set of translated DEFT, we test the translation abilities of our chosen LLMs with a BLEU score and two metrics based on embeddings.

The BLEU score [28] is a widely used metric in machine translation. The mechanism of this metric is to calculate the overlap between n-grams of the gold translation and the translation provided by the system. The known disadvantage of this approach is that lexical overlap does not guarantee meaning preservation. To estimate to which the sense is preserved, we use embedding-based metrics as they operate on a semantic level.

We use the LaBSE model [29] as a cross-lingual encoder for texts, as it encodes semantically close texts in different languages to close points in one embedding space. This allows us to measure the translation quality in two ways.

First, we calculate the mean distance between the corresponding English text and its gold Russian translation, then we do it in the same way between the English text and the translated one. Next, we compare two means by substituting the latter for the former. The closer to the zero metric is, the more likely that model-translated text conveys the same meaning as the gold-translated text. We call this metric Parallel Comparison.

Second, resembling the BLEU approach, we compare the mean distances between gold Russian text embeddings and translated ones. If the BLEU operates on a lexical level, this metric does this on a semantic level and relies on the abovementioned property of cross-lingual encoders. We name this metric BLEU-like.

We reuse the best prompt from the annotation transferring task, only changing the task description. As we will show in the result section, they perform similarly, so we provided the one that we use in Appendix C.

4. Results

4.1. Annotation Transferring

As mentioned earlier, we try to transfer NER annotation by asking LLMs to write triplets in Russian tasks according to the original English spans. However, the LLMs failed to provide a good result in every setting of prompts and even when using ChatGPT-4. The best result we managed to achieve is only 2 out of 20 testing subset examples with this model. We notice several issues related to this failure:

- Wrong index determination: The model did not use the provided indices for exact text extraction. It tried to figure out by itself which text should be extracted instead.
- Span length mismatch: When the model tried to follow the provided indices, the extracted text length did not match with actual span text.
- Ineffectiveness of the task correction: The efforts to explain the task more precisely did not bring valuable improvements.
- Failures of the self-generated prompts: We try to generate prompts by the model itself after providing the detailed task description. However, it also did not help the model to get better at using indices.

After changing the task formulation to extract a substring directly, in a short time, we found a 2-shot prompt that can correctly solve 18 out of 20 testing tasks with ChatGPT3.5-turbo and Llama3.1-8B (we did not test other LLMs, as they were added in the late stage of the work). We chose to use this prompt in the next experiments. The text of the prompt can be found in Appendix A.

While testing the Llama on the gold set, we noticed two things:

- The model tends to break the data format or even send the code, as we would ask her to write a function to transfer the spans by the available index. Generally, this can be fixed just by resending the request.
- The model makes way more unmatchable mistakes. Analysis shows that the source of them is a rich Russian morphology. Sometimes the model corrects the mistakes of the Russian text that was automatically translated. To address this issue, we apply fuzzy matching as a post-process for Llama3.1.

The metric results on the whole gold set for this prompt and the models are provided in Table 1. As seen, the deepseek-chat-v3-0324+fs model shows the best results, where only 0.84% of completely mismatched spans are in common. The comparable results show other LLMs except for gpt-4.1-nano and llama-3.1-8b-instruct. As for the latter, we see that it has a much bigger percentage of mismatched spans than the other LLMs and only a little more than half of exact matches. However, the post-processing fuzzy search can effectively reduce the number of mismatches at the cost of a nonexact mismatch rising. We hypothesize that fine-tuning the post-processing could further improve the result. Generally, we think that if one makes the instruction tuning of the Llama, it could show a much stronger result. It can certainly be found in cases where this strategy makes sense, considering the much lower inference cost of small models.

Lastly, we would like to compare the transferring task in our first view and those we ended up with. The first one that implies using indices consists of the next nonexhaustive cognitive steps: in the English text, find the symbols according to the start index and the end symbol, select symbols in between, translate the resulting substring into another language, locate in the text in another language the corresponding text, and determine the indices in such a way that the final substring will be necessary and sufficient. On the other hand, in the variant where only substrings are used, the steps are similar except there are no steps with the index-related operations, so we can hypothesize that the second task is easier from a cognitive perspective. The LLM fails on the first task, given that it is not bad at math [30], so including counting objects, we might say that the task complexity is accounted for, as the number of cognitive steps is crucial for the LLM to complete the task successfully.

Table 1. The results of the NER transferring. We omit some model details to narrow down the table. fs means "fuzzy search".

Model	Exact Match (%)	Wider Match (%)	Narrower Match (%)	Mismatched (%)	Spans Checked (%)
llama-3.1-8b-instruct	66.57	7.02	4.72	12.42	90.45
llama-3.1-8b-instruct+fs	71.18	8.31	7.08	4.16	90.45
gpt-3.5-turbo	90.62	4.27	1.97	3.09	99.72
gpt-3.5-turbo+fs	91.63	4.78	2.08	1.46	99.72
gpt-4o-mini	86.91	10.00	0.96	2.13	**99.78**
gpt-4o-mini+fs	87.42	10.17	1.12	1.29	**99.78**
gpt-4.1-nano	69.72	9.44	2.36	12.92	93.76
gpt-4.1-nano+fs	75.39	11.18	3.20	4.66	93.76
gpt-4.1-mini	92.42	5.28	1.18	1.12	**99.78**
gpt-4.1-mini+fs	92.64	5.28	1.24	**0.84**	**99.78**
deepseek-chat-v3-0324	94.04	3.65	1.29	1.01	99.78
deepseek-chat-v3-0324+fs	**94.21**	3.65	1.29	**0.84**	**99.78**
deepseek-r1	91.80	5.11	1.69	1.40	**99.78**
deepseek-r1+fs	91.97	5.11	1.69	1.24	**99.78**
qwen-2.5-72b-instruct	89.61	6.40	1.01	2.25	98.99
qwen-2.5-72b-instruct+fs	90.28	6.52	1.07	1.40	98.99

Note: Bold values indicate the highest scores in the Exact Match column, the lowest values in the Mismatched column, and the highest percentages in the Spans Checked column.

4.2. Text Translation

The results for the two prompts that we used are shown in Table 2. We tested them only on gpt-3.5-turbo and llama-3.1-8b at the earlier stage of the research. It is clear that prompts perform very closely in terms of all metrics. Regarding the BLEU score for gpt-3.5-turbo, results for this task are notably higher than the result obtained in work [6] where the ChatGPT obtains a BLEU score of around 45 points on the Eng–Rus pair. We also manually checked several dozen random examples to ensure the sanity of the translation. The text of the two prompts can be found in Appendixes B and C. At the late stage of the work, when we start experimenting with other LLMs, we use prompt 2, as it shows the best performance on gpt-3.5-turbo. We are aware that one prompt might show different results depending on the LLM, but our results in Table 2 show that this difference is small despite the difference in LLM scale. As a result, after comparing other LLMs, gpt-4o-mini and deepseek-chat-v3-0324 demonstrate the best performance, as shown in Table 3.

Table 2. The results of the text translation.

	LaBSE				**BLEU**	
	BLEU-like		**Parallel Comparision**		**BLEU Score**	
	gpt-3.5-turbo	**llama-3.1-8b**	**gpt-3.5-turbo**	**llama-3.1-8b**	**gpt-3.5-turbo**	**llama-3.1-8b**
Prompt 1	0.2267	0.2806	0.0010	−0.0069	0.5011	0.4076
Prompt 2	0.2288	0.2834	0.0005	−0.0090	0.4993	0.4051

Table 3. The results of the text translation for prompt 2.

	LaBSE		**BLEU**
Model	**BLEU-like**	**Parallel Comparison**	**BLEU Score**
llama-3.1-8b	0.2834	−0.0090	0.4051
gpt-3.5-turbo	0.2288	**0.0005**	0.4993
gpt-4o-mini	**0.2168**	**0.0011**	**0.5277**
gpt-4.1-nano	0.2383	−0.0071	0.4650
gpt-4.1-mini	0.2227	−0.0039	0.4971
deepseek-chat-v3-0324	**0.2140**	**−0.0019**	**0.5254**
deepseek-r1	0.2371	−0.0081	0.4625
qwen-2.5-72b-instruct	0.2468	−0.0066	0.4567

Note: Bold values indicate the best results in each column (highest scores for BLEU-like and BLEU Score, most favorable values for Parallel Comparison), including cases with multiple similar top values.

4.3. Whole Transferring and Model Training

For the complete dataset transfer, we chose the deepseek-chat-v3-0324 model because, while it does not show the best performance in translation, with a little gap from the best-performing model, it shows the best performance in NER annotation transferring, which is the key operation. The time spent on applying the pipeline to the whole DEFT corpus and WCL dataset is presented in Table A2. On the output, we obtain the translated corpus with NER annotation. The statistics of the dataset for two tasks are presented in Figure 1 in the "RUS" legend parts. Next, we use this dataset to train the BERT-like models to establish the baseline for the task of definition detection (Task 1) and named entity recognition (Task 2) for definition and term span detection (for the WCL dataset, the only term span detection). Our base model list is next:

- BERT-base-multilingual [31]—the BERT model trained by Google. A good baseline.
- RuBERT-base-cased [32]—RuBERT pretrained from scratch on Russian texts.
- RoBerta-base (https://huggingface.co/blinoff/roberta-base-russian-v0, accessed on 22 April 2025)—RoBERTa [33] model pretrained on Russian texts.

The results of training on the RuDEFT dataset are presented in Table 4 and training on WCL-Wiki-Ru in Table 5. For the RuDEFT, we can see that the models achieve quite good results in the detection of sentences with definitions. For Task 2, the results are notably weaker, which implies that one may need to verify the correctness of the annotation to improve the recognition quality. It is interesting that RoBERTa shows such bad performance. We also see that the difference between the multilingual BERT and RuBERT is insignificant.

Comparing the manually revisited gold part and LLM-translated test parts, we can see a notable gap between them in definition detection. It suggests that while translating, the LLM induces some sort of bias in the text, which the classification model exploits during training.

On the other hand, there is an interesting behavior in Task 2. First, the gap is much lower in general across all models. Second, the models show better results on the gold part. Third, while the gap between RuBERT and Bert-m is stable for Task 1, for Task 2, Bert-m shows a lower gap than RuBERT. If we take a look at the confusion matrix in Appendix D, we will see that the difference primarily comes from the better recognition of the I-Term in the gold part. Probably, this is related to the fact that NER transfer contains partially wrong spans (narrower or wider), but the models managed to generalize in the right way on the transferred train part, although it also contains erroneous spans. The difference between Bert-m and RuBERT might be explained by the fact that RuBERT, which is completely trained on Russian data, can better recognize the nuances of the language, which makes it more robust to the annotation errors.

We leave a detailed analysis of these two phenomena for future work.

As for the WCL-Wiki-Ru dataset, we can see in Table 5 that all models show excellent results on both tasks. Note that RoBERTa is slightly worse than other models. Such a good result might be explained by the simplicity of the dataset itself. As the definitions come from Wikipedia, they have a well-defined structure. If one looks at the definitions, one notices that they are structured like "The X is/named/etc.", that is, the term is placed at the beginning of the sentence, which is followed by the verb. Also, the dataset contains only a Term entity, which makes the whole task easier.

Table 4. The results of the model training on RuDEFT. Task 1—definition detection, Task 2—term and definition recognition.

	Rubert		Roberta		Bert-m	
Metrics	**Gold**	**Test**	**Gold**	**Test**	**Gold**	**Test**
			Task 1			
Precision	0.72	0.85	0.65	0.83	0.72	0.85
Recall	0.81	0.85	0.70	0.76	0.81	0.83
F1	**0.73**	**0.85**	0.57	0.78	0.72	0.84
			Task 2			
Precision	0.73	0.63	0.68	0.64	0.71	0.66
Recall	0.58	0.55	0.29	0.30	0.52	0.51
F1	**0.64**	**0.59**	0.41	0.41	0.60	0.58

Note: Bold values highlight the maximum F1 scores for each task (Task 1 and Task 2) across all models.

Table 5. The results of the model training on WCL-Wiki-Ru. Task 1—definition detection, Task 2—term and definition recognition.

	Task 1			Task 2		
Metrics	**Rubert**	**Roberta**	**Bert-m**	**Rubert**	**Roberta**	**Bert-m**
Precision	0.96	0.94	0.96	0.85	0.87	0.86
Recall	0.97	0.94	0.96	0.93	0.85	0.91
F1	**0.96**	0.94	**0.96**	**0.89**	0.86	**0.89**

Note: Bold values highlight the maximum F1 scores for each task (Task 1 and Task 2) across all models.

5. Discussion

In this work, we show that the current abilities of LLMs can be used to transfer datasets between languages. While the dataset certainly will remain only "silver"-grade quality, the effort difference between creating such a dataset from scratch and adapting from another language with further verification is huge. Especially when we talk about nontrivial annotations like NER that require the exact positioning in text. The focus of our experiment was the DEFT dataset, which contains a quite challenging task of term and definition recognition. This dataset is important to facilitate the trend analysis tools and models for Russian, which are helpful for the decision-making processes in R&D. We train the BERT-based models on the whole transferred dataset to show that these data actually can be used to train real models to establish a baseline. However, we see that the NER model is weak, which implies verifying the transferred annotation more carefully. In addition, we apply our pipeline to the Wikipedia part of the WCL dataset and show that models show quite good results.

As a side effect, we discover that some tasks might be easier for LLMs to understand than others, and that difference may be significant in terms of output quality. We hypothesize that it depends on several cognitive steps that need to be performed for task solving. We also discovered some shortcomings in the DEFT dataset that should be fixed.

The obvious limitations of our work are that we do not show how the LLM itself would be strong on DEFT tasks. Another one is that we do not compare the quality of supervised translators with LLMs, because it is shown that supervised translators are still better than LLMs.

Author Contributions: Conceptualization, I.S. and I.B.; methodology, D.P., E.T. and I.B.; software, D.P., E.T. and I.B.; validation, D.S.; formal analysis, D.P., E.T. and I.B.; investigation, D.P., E.T. and D.S.; data curation, D.P. and E.T.; writing—original draft preparation, D.P., E.T. and D.S.; writing—review and

editing, I.S. and I.B.; visualization, D.P., E.T. and D.S.; supervision, I.S. and I.B.; project administration, I.S. and I.B. All authors have read and agreed to the published version of the manuscript.

Funding: The experimental study was supported by the Project of the Research Center for Trusted Artificial Intelligence of the Ivannikov Institute for System Programming of the Russian Academy of Sciences.

Data Availability Statement: The data generated in this study are available at Hugging Face: https://huggingface.co/datasets/astromis/ruDEFT (accessed on 22 April 2025), https://huggingface.co/datasets/astromis/WCL_Wiki_Ru (accessed on 22 April 2025). The source code used in this work is publicly available on GitHub: https://github.com/Astromis/research/tree/master/rudeft (accessed on 22 April 2025) . These links ensure open access to the data and code in accordance with MDPI's research data policies. The full policy is available at https://www.mdpi.com/ethics (accessed on 22 April 2025).

Acknowledgments: We would like to thank the reviewers for their valuable comments that helped us to improve our paper.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Prompt for Annotation Transferring

Given a JSON object, find the exact corresponding text in the Russian translation for each English span and store the results in a new field called spans_rus. The input JSON object contains the following fields:

- text: English source text.
- text_rus: Russian translated text.
- spans: A list of spans, each containing: 1. The start index in the English text; 2. The end index in the English text; 3. The label; 4. The ID; 5. The portion of the English text that was extracted using the start and end indices.
- And other fields.

Your task is to: For each span, locate the exact corresponding Russian text in text_rus that matches the exact wording of the English span (the 5th element in each span) in meaning.

Important:

- Do not modify or correct the form, word order, or any grammatical aspects of the Russian text — it must be extracted exactly as it appears in text_rus, including word endings, grammatical cases, punctuation, punctuation marks, and spacing.
- Record the matched Russian text as a new list in a new field spans_rus, where each item is also a list containing the same label and ID as in the English span, and the matched Russian text.
- For each span in spans, one must obtain a span in spans_rus.

No explanation, just output the updated JSON.

Appendix B. Prompt 1 for Translation

Given a JSON object, write an accurate translation into Russian for the original English sentence and save the results in a new field named text_rus. The input JSON object contains the following fields:

- id: Unique ID of sentence.
- text: English source text.

Your task is to: For each text in English (text) write its exact translation into Russian in a scientific lexical style and save the results in a new field named text_rus.

Important:

- Write down the corresponding translated Russian text in the form of a new text_rus field.
- For English text in text, one should definitely obtain the Russian text in text_rus.

No explanation, just output the updated JSON.

Appendix C. Prompt 2 for Translation

Given a JSON object, write an accurate translation into Russian for the original English sentence and save the results in a new field named text_rus. The input JSON object contains the following fields:

- id: Unique ID of sentence.
- text: English source text.

Your task is to: For each text in English write its exact translation into Russian taking into account the style of the sentence and its scientific significance (for example, medical, historical, etc.) and save the results in a new field named text_rus.

Important:

- Write down the corresponding translated Russian text in the form of a new text_rus field.
- For English text in text, one should definitely obtain the Russian text in text_rus.

No explanation, just output the updated JSON.

Appendix D. Confusion Matrices

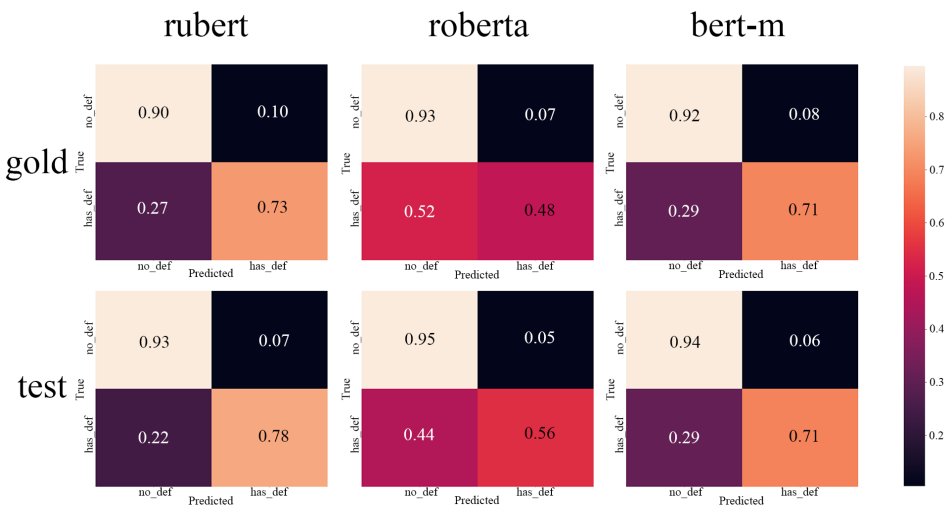

Figure A1. Confusion matrices for Task 1 on the RuDEFT dataset.

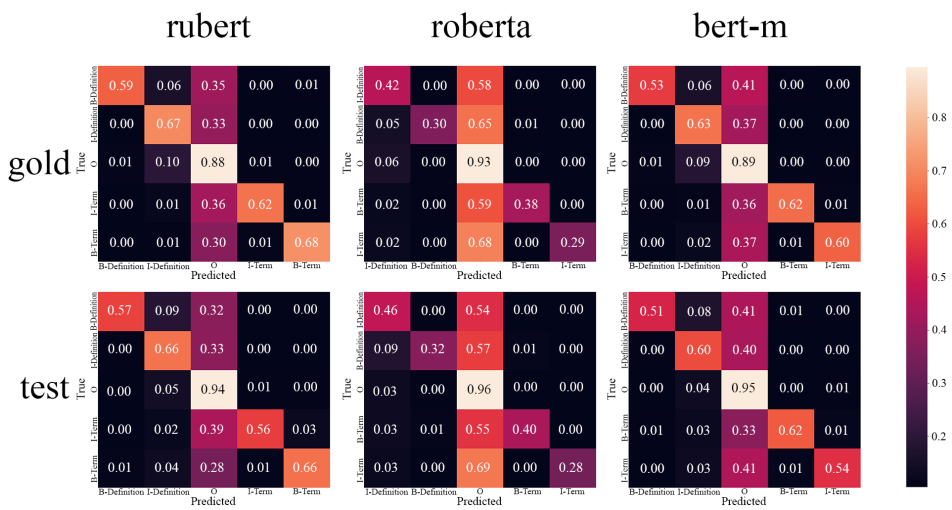

Figure A2. Confusion matrices for Task 2 on the RuDEFT dataset.

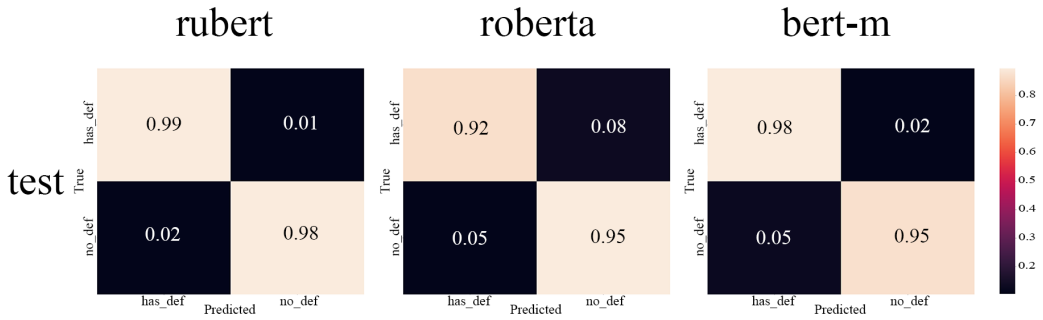

Figure A3. Confusion matrices for Task 1 on the Wikipedia part of the WCL dataset.

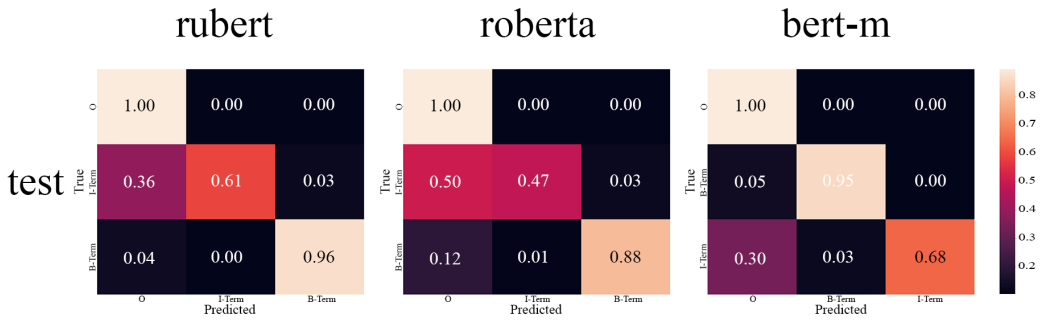

Figure A4. Confusion matrices for Task 2 on the Wikipedia part of the WCL dataset.

Appendix E. LLM Usage: Time and Cost Statistics Across Tasks

Table A1. Comparison of model metrics. CAPS/ex.—CAPS per example (CAPS—the internal currency of the bothub.chat service), USD/ex.—Dollars per example, Time/ex.—Time per example in seconds.

	Text Translation			NER Transferring		
Model	**CAPS/ex.**	**USD/ex.**	**Time/ex. (s)**	**CAPS/ex.**	**USD/ex.**	**Time/ex. (s)**
llama-3.1-8b-instruct	14	0.00002	2.94	121	0.00021	16.8
gpt-3.5-turbo	410	0.00072	1.28	1967	0.00347	9.6
gpt-4o-mini	119	0.00021	0.03	627	0.00111	9.6
gpt-4.1-nano	79	0.00014	0.02	410	0.00072	6
gpt-4.1-mini	312	0.00055	0.02	1595	0.00282	7.2
deepseek-chat-v3-0324	222	0.00039	0.14	1159	0.00205	36
deepseek-r1	1590	0.00281	36.52	5132	0.00906	68.4
qwen-2.5-72b-instruct	111	0.00020	0.05	582	0.00103	28.8

Table A2. Resources spent on pipeline processing of datasets.

	Text Translation			NER Transferring		
Dataset	**CAPS**	**Dollars (USD)**	**Time (min)**	**CAPS**	**Dollars (USD)**	**Time (min)**
RuDEFT	6,497,571	11.5	503	13,125,904	23.2	608
WCL-Wiki-Ru	1,030,348	1.8	49	2,031,111	3.6	98

Appendix F. Software and Dependencies

All experiments and data processing were carried out using Python 3.11. The following libraries, frameworks, and external APIs were installed with the specified versions to ensure full reproducibility:

- langchain==0.3.1
- pydantic==2.9.2
- fuzzysearch==0.7.3
- datasets==2.21.0 (Hugging Face)
- sentence_transformers==3.1.1
- scipy==1.14.0
- evaluate==0.4.3
- label-studio==1.12.1

For a complete list of all dependencies and their exact versions, please refer to the requirements.txt file in the project repository on GitHub.

In addition, we leveraged external services and APIs for specific tasks:

- Google Translate API for automated text translation.
- bothub.chat API as a unified proxy for accessing multiple model families.

References

1. Lobanova, P.; Bakhtin, P.; Sergienko, Y. Identifying and Visualizing Trends in Science, Technology, and Innovation Using SciBERT. *IEEE Trans. Eng. Manag.* **2024**, *71*, 11898–11906. [CrossRef]
2. Lobanova, P.A.; Kuzminov, I.F.; Karatetskaia, E.Y.; Sabidaeva, E.A.; Anpilogov, V.V. Trend Detection Using NLP as a Mechanism of Decision Support. *Sci. Tech. Inf. Process.* **2023**, 88–98. [CrossRef]
3. Devyatkin, D.A.; Nechaeva, E.; Suvorov, R.E.; Tikhomirov, I. Mapping the Research Landscape of Agricultural Sciences. *Foresight Sti Gov.* **2018**, *12*, 69–78. [CrossRef]
4. Spala, S.; Miller, N.A.; Yang, Y.; Dernoncourt, F.; Dockhorn, C. DEFT: A corpus for definition extraction in free- and semi-structured text. In Proceedings of the 13th Linguistic Annotation Workshop, Florence, Italy, 1 August 2019.
5. Spala, S.; Miller, N.A.; Dernoncourt, F.; Dockhorn, C. SemEval-2020 Task 6: Definition Extraction from Free Text with the DEFT Corpus. *arXiv* **2020**, arXiv:2008.13694.
6. Zhu, W.; Liu, H.; Dong, Q.; Xu, J.; Kong, L.; Chen, J.; Li, L.; Huang, S. Multilingual Machine Translation with Large Language Models: Empirical Results and Analysis. *arXiv* **2023**, arXiv:2304.04675.
7. Zhou, W.; Zhang, S.; Gu, Y.; Chen, M.; Poon, H. UniversalNER: Targeted Distillation from Large Language Models for Open Named Entity Recognition. *arXiv* **2023**, arXiv:2308.03279.
8. Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. The Llama 3 Herd of Models. *arXiv* **2024**, arXiv:2407.21783.
9. Klavans, J.L.; Muresan, S. Evaluation of the DEFINDER system for fully automatic glossary construction. In Proceedings of the AMIA Symposium, Washington, DC, USA, 3–7 November 2001; pp. 324–328.
10. Navigli, R.; Velardi, P. Learning Word-Class Lattices for Definition and Hypernym Extraction. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, 13 July 2010.
11. Frantzi, K.T.; Ananiadou, S. The C-value/NC-value domain-independent method for multi-word term extraction. *J. Nat. Lang. Process.* **1999**, *6*, 145–179. [CrossRef]
12. Sun, Y.; Zhuge, H. Discovering Patterns of Definitions and Methods from Scientific Documents. *arXiv* **2023**, arXiv:2307.01216.
13. Collard, J.; de Paiva, V.C.V.; Subrahmanian, E. Parmesan: Mathematical concept extraction for education. *arXiv* **2023**, arXiv:2307.06699.
14. Kang, D.; Head, A.; Sidhu, R.; Lo, K.; Weld, D.S.; Hearst, M.A. Document-Level Definition Detection in Scholarly Documents: Existing Models, Error Analyses, and Future Directions. *arXiv* **2020**, arXiv:2010.05129.
15. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
16. Vaswani, A.; Shazeer, N.M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
17. Navigli, R.; Velardi, P.; Ruiz-Martínez, J.M. An Annotated Dataset for Extracting Definitions and Hypernyms from the Web. In Proceedings of the International Conference on Language Resources and Evaluation, Valletta, Malta, 17–23 May 2010.
18. Martin-Boyle, A.; Head, A.; Lo, K.; Sidhu, R.; Hearst, M.A.; Kang, D. Complex Mathematical Symbol Definition Structures: A Dataset and Model for Coordination Resolution in Definition Extraction. *arXiv* **2023**, arXiv:2305.14660.
19. Bruches, E.P.; Pauls, A.; Batura, T.; Isachenko, V. Entity Recognition and Relation Extraction from Scientific and Technical Texts in Russian. In Proceedings of the 2020 Science and Artificial Intelligence conference (S.A.I.ence) Novosibirsk, Russia, 14–15 November 2020; pp. 41–45.

20. He, X.; Lin, Z.W.; Gong, Y.; Jin, A.; Zhang, H.; Lin, C.; Jiao, J.; Yiu, S.M.; Duan, N.; Chen, W. AnnoLLM: Making Large Language Models to Be Better Crowdsourced Annotators. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Mexico City, Mexico, 16–21 June 2024.
21. Alizadeh, M.; Kubli, M.; Samei, Z.; Dehghani, S.; Zahedivafa, M.H.; Bermeo, J.D.; Korobeynikova, M.; Gilardi, F. Open-Source LLMs for Text Annotation: A Practical Guide for Model Setting and Fine-Tuning. *J. Comput. Soc. Sci.* **2023**, *8*, 17. [CrossRef] [PubMed]
22. Tian, Y.; Ravichander, A.; Qin, L.; Bras, R.L.; Marjieh, R.; Peng, N.; Choi, Y.; Griffiths, T.L.; Brahman, F. MacGyver: Are Large Language Models Creative Problem Solvers? In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Mexico City, Mexico, 16–21 June 2024.
23. Ghanadian, H.; Nejadgholi, I.; Osman, H.A. Socially Aware Synthetic Data Generation for Suicidal Ideation Detection Using Large Language Models. *IEEE Access* **2024**, *12*, 14350–14363. [CrossRef]
24. Frei, J.; Kramer, F. Annotated dataset creation through large language models for non-english medical NLP. *J. Biomed. Inform.* **2023**, *145*, 104478. [CrossRef] [PubMed]
25. Lhoest, Q.; del Moral, A.V.; Jernite, Y.; Thakur, A.; von Platen, P.; Patil, S.; Chaumond, J.; Drame, M.; Plu, J.; Tunstall, L.; et al. Datasets: A Community Library for Natural Language Processing. *arXiv* **2021**, arXiv:2109.02846.
26. Tkachenko, M.; Malyuk, M.; Holmanyuk, A.; Liubimov, N. Label Studio: Data Labeling Software, 2020–2024. Open Source Software. Available online: https://github.com/HumanSignal/label-studio (accessed on 1 April 2025).
27. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; hsin Chi, E.H.; Xia, F.; Le, Q.; Zhou, D. Chain of Thought Prompting Elicits Reasoning in Large Language Models. *arXiv* **2022**, arXiv:2201.11903.
28. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. Bleu: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 6–12 July 2002.
29. Feng, F.; Yang, Y.; Cer, D.M.; Arivazhagan, N.; Wang, W. Language-agnostic BERT Sentence Embedding. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022.
30. Dao, X.Q.; Le, N.B. Investigating the Effectiveness of ChatGPT in Mathematical Reasoning and Problem Solving: Evidence from the Vietnamese National High School Graduation Examination. *arXiv* **2023**, arXiv:2306.06331.
31. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
32. Kuratov, Y.; Arkhipov, M. Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language. *arXiv* **2019**, arXiv:1905.07213.
33. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.