Записки научных
семинаров ПОМИ
Том 546, 2025 г.

D. Kovalevsky, V. Mamedov, S. Stolyarov, S. Ospichev, D. Morozov

FROM PAPERS TO PEERS: LLM-BASED ALGORITHM FOR SELECTING REVIEWERS

Abstract. The rapid growth in the number of annually published scientific papers places a significant burden on the editors of scientific journals and conference organizers. In particular, quickly selecting relevant reviewers becomes difficult. Automation of this process is hampered by the lack of publicly available information on reviewers of already published articles in accordance with the requirements of double-blind peer review. In this paper we aimed to take the first steps toward developing an reviewer recommendation system. Our research focuses on Russian-language scientific articles on mathematics. At the core of our approach lies the comparison of the semantics of the target paper with those of the papers available in the external library. The most similar papers from the library are then aggregated by author, resulting in a list of potential reviewers. This list is subsequently refined through a series of filters. Additionally, we experimented with an extra step: re-ranking the most relevant candidates using a large language model (LLM). To assess the quality of recommendations, we introduced several metrics based on the Universal Decimal Classification (UDC) system, specifically UDC Jaccard similarity and UDC accuracy. The best results were achieved using the E5-multilingual and E5-mistral embedding models. Overall, we were able to achieve a quality higher than 0.88 according to UDC Accuracy@1. The introduction of the LLM-based reranking stage showed mixed results based on preliminary evaluation. While it improved precision and recall metrics at lower k values, human evaluation indicated a preference for the system configuration without reranking. At the same time, the experts' assessments were predominantly positive: most recommendations received ratings of 4 and 5 on a five-point scale.

1. Introduction

The peer review process stands as a fundamental pillar of scientific publishing, ensuring the quality and reliability of academic literature. As the

Key words and phrases: scientific texts, reviewer selection, large language models, text embeddings, recommendation systems.

volume of scientific publications continues to grow exponentially [10, 12], journal editors and conference organizers face mounting challenges, particularly in the critical task of reviewer selection. This can be illustrated by the example of one of the most significant conferences on natural language processing, the Annual Meeting of the Association for Computational Linguistics (ACL). In 2014, 572 long papers were submitted to the conference [20], while in 2024, there were already 4,407 submissions [13].

At the same time, automating the selection of reviewers is a rather difficult process, since editors of a scientific journal cannot have ideal expertise in all areas of knowledge associated with their journal. Selecting candidates based on scientific classifiers, such as the Universal Decimal Classifier (UDC), can be not effective enough, since both the UDC of the reviewed article and the UDC of the reviewers' articles may not be specified narrowly enough. An alternative approach could be based on the semantic analysis of articles in a large scientific library. Such recommendation systems, mainly using machine learning methods, have proven themselves in many areas. However, in the case of selecting reviewers, the task is complicated by the lack of any freely available markup: according to quite clear ethical rules, journals refuse to provide non-anonymized data on reviewers for training recommendation systems.

In our study, we aimed to take the first steps toward developing an reviewer recommendation system. At the core of our approach lies the comparison of the semantics of the target paper with those of the papers available in the external library. The most similar papers from the library are then aggregated by author, resulting in a list of potential reviewers. This list is subsequently refined through a series of filters. Additionally, we experimented with an extra step: re-ranking the most relevant candidates using a large language model (LLM).

We utilized the MathNet[1] scientific library as our data source. Each paper was represented by its title and abstract, while deliberately excluding any author-related information. Since the actual reviewers for these papers are unknown, we employed a proxy metric to evaluate the quality of the recommendations: specifically, we measured whether the model proposed the paper's original author as a potential reviewer. However, recognizing the limitations of this proxy metric (e.g., it may favor models that simply guess the author rather than identifying a suitable field of expertise), we also conducted a draft human evaluation to validate our results.

[1] `https://www.mathnet.ru`

The paper is organized as follows. In Section 2, we provide a concise overview of the relevant research domain. Section 3 outlines the design of our proposed approach. The metrics employed to evaluate the quality of our results are detailed in Section 4. The dataset collected and utilized in our experiments is described in Section 5. Implementation specifics and the experimental setup are enumerated in Section 6. The results of the experiments, along with their analysis, are presented in Section 7. Finally, Section 8 summarizes the key findings and contributions of this work.

2. Related work

Automated reviewer assignment systems for scientific publications have evolved from simple matching techniques to more sophisticated approaches. Research in this area encompasses algorithmic fairness, automated matching systems, language model integration, and optimization of reviewer assignment processes.

For large conferences with established reviewer pools, algorithmic approaches have been developed to address the challenge of distributing reviewers across submissions. Stelmakh et al. [19] proposed the PeerReview4All algorithm, which aims for fair reviewer distribution by maximizing the minimum review quality across papers. This direction has been further explored by Payan et al. [17] using sequential selection mechanisms, and by Leyton-Brown et al. [15], who implemented a system at AAAI 2021 employing mixed-integer programming methods to optimize reviewer assignments.

Expert search systems have evolved over time, beginning with Craswell et al. [6], who introduced P@NOPTIC Expert to identify domain experts based on their published works. The Toronto Paper Matching System (TPMS) [16] established methods for expertise matching using probabilistic graphical models to analyze textual similarity between papers and reviewer profiles. The field has advanced with neural methods, exemplified by Cohan et al. [4], who proposed SPECTER, a system utilizing citation-informed transformers to learn document-level representations.

Several specialized systems for reviewer selection have been developed, each with distinct approaches. ETBLAST evaluated author expertise based

on their position in publication author lists [9]. Jane improved upon ET-BLAST but was limited to medical domains [18]. Peer2ref employed machine learning algorithms and keyword analysis [1]. Comparative evaluation with these systems presents challenges, as some are no longer operational or are restricted to specific domains. De Campos et al. [8] compared information retrieval and machine learning methods for academic expert finding, though their assumption of single authorship limits applicability to multi-authored publications.

The integration of large language models (LLM) represents a recent development in this field. Tyser et al. [21] identified issues when using LLMs in review processes, while Goldberg et al. [11] conducted experiments at NeurIPS 2024 where LLMs served as checklist assistants. D'Arcy et al. [7] introduced the ARIES dataset, demonstrating challenges computational models face when connecting reviewer comments to paper edits.

Current approaches face limitations in the representation of expertise, evaluation methodologies, and system transparency. The field has evolved from addressing mathematical optimization problems to developing integrated systems, and large language models (LLMs) demonstrate potential for supporting human reviewers despite existing limitations that prevent fully automated reviewer selection.

3. Our approach

Building upon the challenges outlined in the introduction, we developed a two-stage pipeline for reviewer recommendation. Our approach addresses the lack of ground-truth reviewer data by leveraging semantic similarity between papers as a proxy for expertise matching. The first stage employs neural embeddings [3] to process the input paper's metadata (title and abstract), identifying similar papers in the database and aggregating these results to generate an initial pool of potential reviewers. The second stage refines these candidates using a large language model that evaluates deeper aspects of expertise alignment, including research trajectory and methodological experience. This architecture supports flexible filtering based on subject categories and publication activity, while allowing us to measure the impact of LLM-based re-ranking on recommendation quality. The following sections detail each component of our pipeline.

3.1. Main Stage: Candidate Selection. The main stage implements a candidate selection process through semantic embedding-based search.

We vectorize the article's title and abstract using one of five embedding models we evaluated: E5-multilang [23], E5-mistral [22], RuBERT-tiny2 [5], SciBERT [2], and Sci-RUS-tiny3.1 [14]. Each model offers different dimensional representations optimized for various language contexts.

After embedding generation, we perform nearest neighbor search to identify the most similar articles from our pre-vectorized database. These articles are ordered by descending cosine similarity with the query article, and the top-100 results are retained. We then aggregate these results at the author level by computing mean similarity scores across all articles by each author, which provides a balanced representation of expertise while penalizing one-time contributions and rewarding consistent domain expertise. Finally, we apply additional filtering criteria, requiring authors to appear in at least $N_{\min}$ articles among the top-K search results and to have published at least $P_{\min}$ articles within the past Y years. Through preliminary experiments aimed at balancing recommendation quality and candidate pool size, we established the following parameter values: $N_{\min} = 2$, $K = 100$, $P_{\min} = 5$, and $Y = 10$. This process yields a ranked list of candidate reviewers with associated mean similarity scores reflecting their expertise alignment with the submitted paper.

These values are hyperparameters of our system, which we selected based on preliminary experiments to balance between recommendation quality and candidate pool size.

3.2. Additional Stage: Re-ranking. To further refine our recommendations, we implemented a re-ranking stage utilizing a large language model. Our preliminary experiments evaluated various open-source LLMs from the Llama 3.1 family with parameter sizes 8B, 70B, 405B. We found that the 8B model lacked sufficient capacity for this complex evaluation task, while the 405B model was computationally excessive. Ultimately, we selected the 70B parameter model as it provides an optimal balance between computational efficiency and evaluation quality. The re-ranking methodology employs few-shot prompting, where the model analyzes comprehensive information about each candidate reviewer alongside their relevant publications and the target paper's details (see Appendix A). The model then evaluates candidates on a 1-10 scale across multiple dimensions of expertise alignment. To enhance computational efficiency without compromising evaluation quality, we process multiple candidates simultaneously within individual prompts, significantly reducing the overall processing requirements while maintaining robust assessment capabilities.

4. Metrics

To evaluate our system's performance, we employed metrics that assess both article-level thematic alignment and author-level expertise matching. These metrics correspond directly to the evaluation results presented in subsequent sections.

For article-level assessment, we utilize *Article UDC Precision@K*, which measures the proportion of top-K closest articles with the same UDC code. This is calculated as the ratio of articles with overlapping UDC codes to the total number of similar articles at position K:

$$\text{Article UDC Precision@K} = \frac{\sum_{i=1}^{K} \mathbb{I}(\text{UDC}_i \cap \text{UDC}_{\text{article}} \neq \emptyset)}{K}$$

For author-level evaluation, we employ several complementary metrics. *Author Recall@K* quantifies the proportion of actual paper authors captured within the top K recommendations:

$$\text{Author Recall@K} = \frac{\text{Number of actual authors in top K recommendations}}{\text{Total number of paper authors}}$$

Author Precision@K measures the proportion of actual paper authors among the top K recommended reviewers:

$$\text{Author Precision@K} = \frac{\text{Number of actual authors in top K recommendations}}{K}$$

Additionally, *Author UDC Jaccard* quantifies the thematic correspondence between a candidate's research areas and the test article using the Jaccard similarity of truncated UDC code sets (first four characters). This truncation to the main category level allows us to capture broader thematic alignment while reducing the impact of highly specific subcategories that might unnecessarily fragment the similarity assessment:

$$\text{Author UDC Jaccard} = \frac{|\text{UDC}^{(4)}_{\text{author}} \cap \text{UDC}^{(4)}_{\text{article}}|}{|\text{UDC}^{(4)}_{\text{author}} \cup \text{UDC}^{(4)}_{\text{article}}|}$$

where $\text{UDC}^{(4)}_{\text{author}}$ and $\text{UDC}^{(4)}_{\text{article}}$ denote the sets of truncated UDC codes for the candidate reviewer and the article, respectively.

Author UDC Accuracy@K calculates the proportion of candidates whose UDC codes match those of the test article at the category level. This metric also uses only the first four characters of the UDC classification code,

which represent the main subject category, providing a more generalizable measure of disciplinary alignment:

$$\text{Author UDC Accuracy@K} = \frac{\sum_{i=1}^{K} \mathbb{I}(\text{UDC}_i^{(4)} \cap \text{UDC}_{\text{article}}^{(4)} \neq \emptyset)}{K}$$

where $\text{UDC}_i^{(4)}$ and $\text{UDC}_{\text{article}}^{(4)}$ denote the sets of truncated UDC codes for the candidate reviewer i and the article, respectively.

We evaluated these metrics systematically across different embedding models and assessed the impact of the LLM-based reranking stage on overall recommendation quality.

5. Data

The evaluation utilized the MathNet.ru library, a collection of Russian mathematical research papers. Our corpus derived from this library contains 328,286 scientific articles dating from the 19th century to 2024, with contributions from 167,277 unique authors. UDC classifications are available for 61.8% of the papers (202,925 articles), while the remaining entries may represent conference abstracts, preprints, or other academic materials without formal classification. Figures 1 and 2 illustrate the distribution of authors per paper and publication trends over time, respectively.

Analysis of the most frequent UDC codes demonstrates the corpus's mathematical focus, with predominant classifications including differential equations (517.9, 6,793 articles), function theory (517.5, 3,871 articles), mathematical physics equations (517.958, 3,509 articles), and mathematical statistics (519.6, 3,202 articles). This distribution reflects the specialized nature of the MathNet.ru repository, which emphasizes theoretical mathematics and its applications. Statistical analysis of the UDC distribution reveals a critical class imbalance, with a Gini coefficient of 0.9042 indicating extreme disparity. The top 20% of documents are concentrated in just one class (less than 0.01% of all classes), while the bottom 20% span 20,579 classes (96.39% of all classes). The ratio between the most and least represented classes reaches 140,863, with a median of just one document per class. This severe imbalance is particularly relevant for classification methodologies and result interpretation, as standard machine learning algorithms may exhibit systematic bias toward dominant classes.

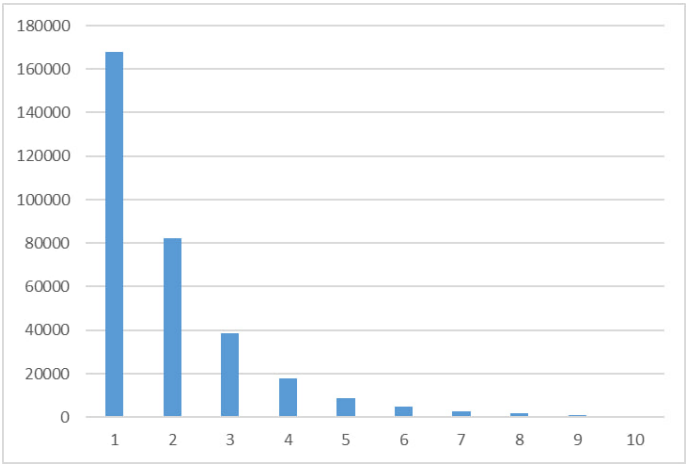

Figure 1. Number of authors per article

6. Experimental Setup

The experiments were conducted with the following configuration parameters: maximum number of authors considered per paper (`max_authors = 100`), maximum number of articles retrieved in the first stage (`max_articles = 100`), test interval spanning 2023–2025, and retrieval interval covering 2012–2022. This temporal separation was deliberately chosen to model real-world scenarios where reviewer recommendations must be made for new papers based on historical publication data.

For encoder evaluation, we compared several models using identical preprocessing and evaluation pipelines, including models described in Section 3.1. The LLM reranking experiments employed a Llama 3.1 70B parameter model selected after preliminary testing with various open-source LLMs.

Our experimental methodology included an evaluation of a standardized prompt for both the embedding and reranking stages.

For the embedding stage, we utilized a consistent input format that included the paper's title and abstract in Russian (as shown in Appendix A). The E5 family of models, which supports instruction-based prompting, was theoretically expected to improve retrieval metrics through the generation of more targeted embeddings for the specific task.

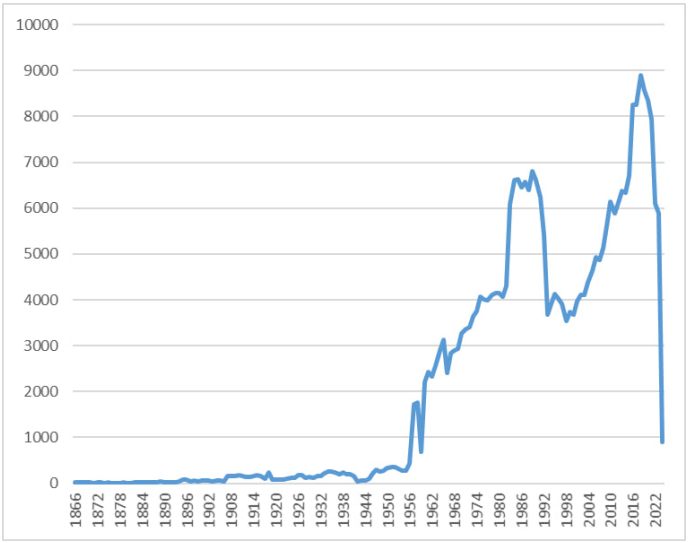

Figure 2. Number of articles published per year

For the reranking stage, we developed a few-shot prompt that instructed the model to evaluate the relevance of potential reviewers based on their research areas and publication history in relation to the target paper. This prompt included explicit scoring criteria on a scale from 1 to 10, focusing on the reviewer's ability to understand the paper's content. The standardized prompt approach ensured consistency across all experiments while effectively capturing the thematic alignment necessary for accurate reviewer matching. The complete prompt templates are documented in Appendix A, providing a reproducible framework for the system.

For the human evaluations, we utilized the E5-multilingual model as our embedder. Quantitative analysis showed that larger LLM encoders did not provide significant improvements in performance metrics despite requiring more computational resources.

6.1. Computational Resources. The experiments were conducted on a server equipped with an NVIDIA RTX 4090 GPU. For inference, we utilized VLLM with CPU offloading capabilities, particularly for the E5-mistral encoder and the Llama 3.1 70B reranker model. All models were

run with BF16 precision to optimize the balance between computational efficiency and numerical accuracy.

The vectorization of the entire MathNet.ru database, comprising over 328,000 scientific articles, required computational time up to approximately 6 hours, depending on the encoder model used. This one-time preprocessing step generated persistent embeddings that could be reused across multiple experimental configurations. Since our approach does not involve fine-tuning or training new models, the majority of computational resources were allocated to real-time inference during the evaluation phase.

For the first-stage retrieval using various encoder models, the average processing time per query was under 1 seconds when using pre-computed embeddings and an optimized nearest neighbor search implementation. The second-stage reranking with the 70B parameter LLM required more substantial resources, with an average processing time of approximately 3-5 seconds per batch of candidate reviewers when using VLLM's optimized inference pipeline.

This computational profile makes the system practical for real-world deployment scenarios, where reviewer recommendations need to be generated within reasonable timeframes for conference submission systems or journal editorial workflows.

7. Results and discussion

7.1. Encoder Model Comparison. Table 1 presents the comparative performance of encoder models across various metrics. For article-level metrics, E5-mistral showed the highest precision values at 66.1% for k=1, with E5-multilang following at 59.5%. The sci-rus-tiny3.1 model produced results comparable to E5-multilang, while rubert-tiny2 and scibert achieved lower precision values, with scibert recording only 26.7% for k=1.

Regarding author-level metrics, E5-multilang exhibited better performance on recall and precision, with E5-mistral showing the second-best results. At k=1, E5-multilang reached 15.8% precision, indicating that for approximately one in six papers, the top recommended reviewer was an author of the paper.

For UDC-based metrics, E5-mistral outperformed E5-multilang, particularly for the Jaccard similarity measure. The UDC accuracy metric showed values of approximately 88-89% across all models at k=1, suggesting that the top recommended reviewer generally possessed expertise in the same broad subject area as the paper.

Table 1. Performance Metrics Across Different Encoder Models

Metric	@k	multilang	mistral	rubert-tiny2	scibert	sci-rus-tiny3.1
Article UDC Precision	1	0.595	0.661	0.467	0.267	0.589
	3	0.551	0.622	0.410	0.217	0.543
	5	0.521	0.599	0.383	0.197	0.515
	10	0.485	0.568	0.341	0.173	0.480
	20	0.441	0.531	0.310	0.155	0.444
	50	0.388	0.482	0.269	0.133	0.398
	100	0.347	0.448	0.245	0.120	0.372
Author Recall	1	0.010	0.009	0.007	0.004	0.008
	3	0.024	0.019	0.015	0.007	0.018
	5	0.030	0.024	0.019	0.008	0.024
	10	0.039	0.031	0.024	0.010	0.033
	20	0.047	0.037	0.029	0.013	0.043
	50	0.054	0.043	0.038	0.017	0.054
	100	0.059	0.047	0.043	0.021	0.059
Author Precision	1	0.158	0.135	0.112	0.061	0.128
	3	0.132	0.107	0.085	0.041	0.103
	5	0.107	0.085	0.068	0.032	0.087
	10	0.074	0.058	0.047	0.021	0.064
	20	0.047	0.036	0.030	0.013	0.043
	50	0.023	0.017	0.016	0.008	0.023
	100	0.013	0.009	0.009	0.005	0.013
Author UDC Jaccard	1	0.654	0.654	0.612	0.566	0.652
	3	0.481	0.513	0.421	0.341	0.477
	5	0.426	0.457	0.354	0.264	0.422
	10	0.364	0.401	0.287	0.186	0.364
	20	0.318	0.358	0.235	0.141	0.318
	50	0.272	0.318	0.195	0.107	0.278
	100	0.244	0.292	0.175	0.089	0.258
Author UDC Accuracy	1	0.888	0.885	0.886	0.896	0.884
	3	0.674	0.716	0.629	0.556	0.663
	5	0.603	0.647	0.536	0.436	0.594
	10	0.524	0.576	0.439	0.316	0.521
	20	0.466	0.523	0.367	0.241	0.464
	50	0.410	0.477	0.314	0.186	0.418
	100	0.372	0.445	0.285	0.158	0.391

The relatively poor performance of scibert is not unexpected given its training exclusively on English scientific literature. Its inclusion in our evaluation serves as a validation point, confirming that models trained on multilingual corpora or specifically on Russian scientific texts perform better for our task. This finding is logical considering that the MathNet.ru database contains predominantly Russian-language mathematical papers.

Our analysis reveals that model size does not necessarily correlate with performance in this specialized task. E5-multilang (560M parameters) outperforms larger models on several metrics, particularly in author identification. This suggests that domain adaptation and training objectives may

Table 2. Author Metrics with Different Prompting Strategies

Metric	@k	Without prompts		Queries + prompt		DB + queries + prompts	
		multilang	mistral	multilang	mistral	multilang	mistral
Recall	1	0.010	0.009	0.010	0.008	0.009	0.009
	3	0.024	0.019	0.022	0.018	0.022	0.022
	5	0.030	0.024	0.028	0.023	0.029	0.028
	10	0.039	0.031	0.037	0.030	0.037	0.037
	20	0.047	0.037	0.044	0.036	0.045	0.045
	50	0.054	0.043	0.053	0.042	0.053	0.054
	100	0.059	0.047	0.058	0.046	0.057	0.059
Precision	1	0.158	0.135	0.152	0.128	0.150	0.142
	3	0.132	0.107	0.124	0.101	0.127	0.121
	5	0.107	0.085	0.101	0.081	0.104	0.098
	10	0.074	0.058	0.071	0.056	0.072	0.069
	20	0.047	0.036	0.045	0.035	0.045	0.045
	50	0.023	0.017	0.022	0.017	0.022	0.022
	100	0.013	0.009	0.012	0.009	0.012	0.012

be more significant factors than model scale for scientific reviewer recommendation systems. The strong performance of sci-rus-tiny3.1, despite its smaller size, further supports this conclusion, as it was specifically fine-tuned on Russian scientific literature.

The quantitative analysis indicates that E5-multilang and E5-mistral are the most suitable encoder models for the reviewer recommendation system, with E5-multilang showing stronger performance in author identification metrics and E5-mistral in thematic alignment metrics.

7.2. Impact of Prompt Engineering in Embedder Models. The influence of prompt engineering on retrieval performance was examined by comparing three configurations: no prompts, prompts for queries only, and prompts for both database and queries, with results presented in Table 2.

Analysis indicates that prompt engineering has a limited effect on performance metrics, with base models without prompts generally exhibiting higher precision values. For E5-multilang, precision@1 decreases from 15.8% without prompts to 15.2% with query prompts and 15.0% with both database and query prompts. Article UDC precision@1 similarly decreases from 59.5% to 59.0% and 58.7%, respectively.

For E5-mistral, the effect of prompting varies across metrics. Author precision@1 decreases from 13.5% to 12.8% with query prompts, but increases to 14.2% when both database and query prompts are applied. Article UDC precision@1 decreases from 66.1% without prompts to 65.2% with query prompts and 58.0% with both prompts.

Table 3. Performance Metrics with and without Reranking.

Metric	@k	multilang		mistral	
		base	rerank	base	rerank
Precision	1	0.110	0.200	0.110	0.200
	3	0.097	0.150	0.083	0.147
	5	0.072	0.126	0.070	0.122
	10	0.052	0.075	0.057	0.075
	20	0.035	0.041	0.037	0.044
	50	0.018	0.018	0.019	0.019
Recall	1	0.008	0.015	0.007	0.012
	3	0.017	0.028	0.014	0.023
	5	0.020	0.036	0.021	0.033
	10	0.030	0.041	0.032	0.039
	20	0.037	0.043	0.038	0.043
	50	0.046	0.046	0.046	0.046
UDC Jaccard	1	0.725	0.678	0.690	0.631
	3	0.481	0.432	0.557	0.470
	5	0.409	0.380	0.496	0.455
	10	0.309	0.314	0.431	0.409
	20	0.262	0.271	0.382	0.388
	50	0.212	0.213	0.329	0.330
UDC Accuracy	1	0.893	0.929	0.879	0.886
	3	0.640	0.639	0.727	0.685
	5	0.542	0.571	0.648	0.661
	10	0.433	0.475	0.577	0.596
	20	0.365	0.396	0.523	0.546
	50	0.310	0.312	0.470	0.470

These results suggest that the encoder models function effectively for the scientific domain without additional prompt engineering. The observed decrease in performance with prompts may be attributed to the addition of extraneous text that potentially reduces the specificity of the semantic representation.

7.3. Reranking Contribution. To evaluate our two-stage approach, we examined the effect of the LLM-based reranking stage on recommendation metrics using E5-multilang and E5-mistral models, which demonstrated

the highest performance in initial experiments. Due to computational constraints, this analysis was limited to a subset of 100 documents from the test dataset.

The reranking process employs a few-shot prompting approach that provides the model with candidate reviewer information, similar papers, and the target paper's details (see Table 7 in Appendix A; given the substantial length of the prompt, we have chosen not to reproduce it in full, omitting the descriptions of the authors; an example of an author description is provided in Table 6 in the same Appendix). The model assigns each candidate a relevance score from 1 to 10, considering factors such as research area overlap, methodological expertise, and publication history.

Table 3 presents the quantitative results of applying reranking to various evaluation metrics. The reranking stage improves precision and recall metrics, with notable effects at lower K values. For both E5-multilang and E5-mistral, precision@1 increases from 11.0% to 20.0%. For recall@1, E5-multilang shows an increase from 0.8% to 1.5%, while E5-mistral exhibits an increase from 0.7% to 1.2%.

The UDC Jaccard similarity demonstrates a decrease at the top positions with reranking (from 0.725 to 0.678 for E5-multilang at k=1), whereas the UDC accuracy metrics increase, with E5-multilang UDC accuracy@1 changing from 0.893 to 0.929. This pattern suggests that the reranker may prioritize candidates with specific thematic alignment rather than those with broader expertise profiles.

The observed changes in metrics across different K values indicate that the reranker alters the distribution of candidates in the ranking, with effects most evident at lower K values (k=1 to k=5), which is relevant for applications where users primarily examine the highest-ranked recommendations.

7.4. Human Evaluation. During development, expert evaluators assessed the quality of reviewer recommendations. The human evaluation results (Table 4) show differences between system configurations. The version without reranking received higher ratings, with 63.3% of recommendations rated as "Excellent" compared to 21.7% with reranking. Converting ratings to numerical values (Excellent=5, Good=4, Average=3, Poor=2, Very Poor=1), recommendations without reranking averaged 4.36, while those with reranking averaged 3.33.

Table 4. Expert evaluation results for both embedder and reranker approaches.

Rating	Overall	With Reranking	Without Reranking
Excellent	63 (45.3%)	13 (21.7%)	50 (63.3%)
Good	37 (26.6%)	22 (36.7%)	15 (19.0%)
Average	14 (10.1%)	6 (10.0%)	8 (10.1%)
Poor	14 (10.1%)	10 (16.7%)	4 (5.1%)
Very Poor	11 (7.9%)	9 (15.0%)	2 (2.5%)

8. Conclusion

In this paper, we presented an approach to the expert reviewer recommendation problem, utilizing a two-stage architecture that combines embedding-based retrieval with LLM-based reranking. Our system addresses the challenge of matching scientific papers with appropriate reviewers by leveraging text embedding models and contextual understanding capabilities of large language models, with a particular focus on Russian-language mathematical texts.

To assess the quality of recommendations, we introduced several metrics based on the Universal Decimal Classification (UDC) system, specifically UDC Jaccard similarity and UDC accuracy. This decision is due to the fact that information about real reviewers of articles is not available, i.e. it is impossible to compare recommended and real reviewers.

In the aggregate of experiments, the best results were achieved using the E5-multilingual and E5-mistral embedding models. Among the smaller models, the best results were achieved using the sci-rus-tiny3.1 model. Overall, we were able to achieve a quality higher than 0.88 according to UDC Accuracy@1.

The introduction of the LLM-based reranking stage showed mixed results based on preliminary evaluation. While it improved precision and recall metrics at lower k values, particularly for the top recommendations, human evaluation indicated a preference for the system configuration without reranking. This discrepancy between automated metrics and human assessment highlights the complexity of the expert recommendation task and suggests that purely algorithmic evaluation may not fully capture the

nuanced factors that domain experts consider when judging recommendation quality. At the same time, the experts' assessments were predominantly positive: most recommendations received ratings of 4 and 5 on a five-point scale. This confirms the potential applicability of our approach.

The developed system offers a practical solution for academic publishers, conference organizers, and research institutions seeking to streamline the reviewer selection process while maintaining standards of peer review quality, particularly for Russian-language mathematical publications. The observed trade-offs between different system configurations provide valuable insights for future work in this domain. Further research could explore alternative reranking strategies, incorporate additional contextual information about reviewers' expertise, and investigate more sophisticated methods for balancing thematic alignment with broader expertise profiles.

Appendix A. Model Prompts

Table 5. Embedder Input Format

Название статьи:
{title}
Аннотация статьи:
{abstract}

Table 6. Author Description Example

Имя:
—
Горбацевич Владимир Витальевич
—
Похожие работы:
—
- Полиномиальные реализации конечномерных алгебр Ли.
- О локально транзитивных аналитических действиях групп Ли на компактных поверхностях.
- О форме Киллинга на алгебрах Ли.
—
Все работы:
—
- Вычислительные эксперименты с нильпотентными алгебрами Ли
- Дуальные и почти дуальные однородные пространства
- Изоморфность и диффеоморфность полупростых групп Ли
- Некоторые свойства однородных $\mathcal{E}$-многообразий
- Некоторые свойства почти абелевых алгебр Ли
- Об изоморфизме и диффеоморфизме компактных полупростых групп Ли
- О локально транзитивных аналитических действиях групп Ли на компактных поверхностях
- О максимальных расширениях нильпотентных алгебр Ли
- О некоторых классах базисов в конечномерных алгебрах Ли
- О расслоенной структуре компактных однородных пространств
—

Table 7. Reranker Prompt with Few-Shot Examples

Ниже приведён список учёных. Необходимо оценить, насколько учёный квалифицирован, чтобы понять статью.
Для каждого учёного указаны номер, имя, область его работы, а так же похожие на искомую статьи.
После этого будет приведена статья, для неё указаны название и аннотация.
Тебе нужно определить, насколько близки области интересов исследователей к данной статье. В качестве ответа дай номера учёных, а также оценку релевантности. Оценка релевантности - это число от 1 до 10, отражающее насколько хорошо учёный сможет понять статью.
Дай ответ для каждого учёного. Объяснять ответ не нужно.

Пример:
==========
Учёный 1:
=====
{Author 1 description}
=====

Учёный 2.
=====
{Author 2 description}
=====

Учёный 3.
=====
{Author 3 description}
=====

Статья:
Название статьи:
Локальные универсальные алгебры и приводимые представления алгебр Ли

Аннотация статьи:
Рассматриваются представления алгебр Ли над полем характеристики $p > 0$. Вводятся локальные универсальные алгебры и показывается, что изучение приводимых представлений алгебры Ли сводится к изучению представлений ее локальных универсальных алгебр. Это позволяет в некоторых случаях свести изучение приводимых представлений алгебры Ли к изучению представлений коммутативного кольца.

Ответ:
Учёный: 1, Релевантность: 9
Учёный: 2, Релевантность: 3
Учёный: 3, Релевантность: 6

==========

Теперь дай ответ:
{context_str}
Статья:
{query_str}
Ответ:

References

1. M. A. Andrade-Navarro, G. A. Palidwor, and C. Perez-Iratxeta, *Peer2Ref: A Peer-Reviewer Finding Web Tool That Uses Author Disambiguation.* — BioData Mining **5**(1) (2012), 14.
2. I. Beltagy, K. Lo, and A. Cohan, *SciBERT: A Pretrained Language Model for Scientific Text*, in: K. Inui, J. Jiang, V. Ng, and X. Wan (eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, Nov. 2019, pp. 3615–3620.
3. M. Berger, J. Zavrel, and P. Groth, *Effective Distributed Representations for Academic Expert Search*, in: M. K. Chandrasekaran, A. de Waard, G. Feigenblat, D. Freitag, T. Ghosal, E. Hovy, P. Knoth, D. Konopnicki, P. Mayr, R. M. Patton, and M. Shmueli-Scheuer (eds.), Proceedings of the First Workshop on Scholarly Document Processing, Association for Computational Linguistics, Online, Nov. 2020, pp. 56–71.
4. A. Cohan, S. Feldman, I. Beltagy, D. Downey, and D. S. Weld, *SPECTER: Document-Level Representation Learning Using Citation-Informed Transformers*,

in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 2270–2282.
5. Cointegrated, *RuBERT-Tiny2*, 2023. Available: `https://huggingface.co/cointegrated/rubert-tiny2`.
6. N. Craswell, D. Hawking, A.-M. Vercoustre, and P. Wilkins, *P@NOPTIC Expert: Searching for Experts Not Just for Documents*, in: Proceedings of the Australian World Wide Web Conference (AusWeb), 2001.
7. M. D'Arcy, A. Ross, E. Bransom, B. Kuehl, J. Bragg, T. Hope, and D. Downey, *ARIES: A Corpus of Scientific Paper Edits Made in Response to Peer Reviews*, in: L.-W. Ku, A. Martins, and V. Srikumar (eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Bangkok, Thailand, Aug. 2024, pp. 6985–7001.
8. L. M. de Campos, J. M. Fernández-Luna, J. F. Huete, F. J. Ribadas-Pena, and N. Bolanos, *Information Retrieval and Machine Learning Methods for Academic Expert Finding*. — Algorithms **17**(2) (2024).
9. M. Errami, J. D. Wren, J. M. Hicks, and H. R. Garner, *ETBlast: A Web Server to Identify Expert Reviewers, Appropriate Journals and Similar Publications*. — Nucleic Acids Research **35**(suppl. 2) (2007), W12–W15.
10. M. Fire and C. Guestrin, *Over-Optimization of Academic Publishing Metrics: Observing Goodhart's Law in Action*. — GigaScience **8** (2019).
11. A. Goldberg, I. Ullah, T. G. H. Khuong, B. K. Rachmat, Z. Xu, I. Guyon, and N. B. Shah, *Usefulness of LLMs as an Author Checklist Assistant for Scientific Papers: NeurIPS'24 Experiment*, ArXiv preprint arXiv:2411.03417 (2024).
12. M. Gusenbauer, *Google Scholar to Overshadow Them All? Comparing the Sizes of 12 Academic Search Engines and Bibliographic Databases*. — Scientometrics **118** (2019), 177–214.
13. L.-W. Ku, A. Martins, and V. Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Bangkok, Thailand, Aug. 2024.
14. MLSA IAI MSU Lab, *Sci-RuS-Tiny3.1*, 2023. Available: `https://huggingface.co/mlsa-iai-msu-lab/sci-rus-tiny3.1`.
15. K. Leyton-Brown, Mausam, Y. Nandwani, H. Zarkoob, C. Cameron, N. Newman, and D. Raghu, *Matching Papers and Reviewers at Large Conferences*. — Artificial Intelligence **331** (2024), 104119.
16. X. Li and T. Watanabe, *Automatic Paper-to-Reviewer Assignment Based on the Matching Degree of the Reviewers*. — Procedia Computer Science **22** (2013), 633–642. 17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems (KES2013).
17. J. Payan and Y. Zick, *I Will Have Order! Optimizing Orders for Fair Reviewer Assignment*, in: Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS '22), International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2022, pp. 1711–1713.
18. M. J. Schuemie and J. A. Kors, *JANE: Suggesting Journals, Finding Experts*. — Bioinformatics **24**(5) (2008), 727–728.

19. I. Stelmakh, N. Shah, A. Singh, and H. Daumé III, *PeerReview4All: Fair and Accurate Reviewer Assignment in Peer Review*, in: Proceedings of the 30th International Conference on Algorithmic Learning Theory, PMLR, vol. 98, 2019, pp. 828–856.
20. K. Toutanova and H. Wu (eds.), *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Baltimore, Maryland, June 2014.
21. K. Tyser, J. Lee, A. Shporer, M. Udell, D. Te'eni, and I. Drori, *OpenReviewer: Mitigating Challenges in LLM Reviewing*, 2023.
22. L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei, *Improving Text Embeddings with Large Language Models*, in: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2024, pp. 11897–11916.
23. L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei, *Multilingual E5 Text Embeddings: A Technical Report*, ArXiv preprint arXiv:2402.05672 (2024).

Novosibirsk State University
E-mail: d.kovalevskii@g.nsu.ru
E-mail: v.mamedov@g.nsu.ru
E-mail: s.stolyarov@g.nsu.ru
E-mail: s.ospichev@nsu.ru

Novosibirsk State University,
Russian National Corpus
E-mail: morozowdm@gmail.com

Поступило 28 февраля 2025 г.