НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
июль–август 2026 Том 26 № 4 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
July–August 2026 Vol. 26 No 4 http://ntv.ifmo.ru/en/
ISSN 2226-1494 (print) ISSN 2500-0373 (online)

НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

doi: 10.17586/2226-1494-2026-26-4-844-850

Automatic prompt optimization with prompt distillation

Viktor N. Zhuravlev[1], Artur R. Khairullin[2], Ernest A. Dyagin[3], Alyona N. Sitkina[4], Nikita I. Kulin[5]✉

[1,2,3,4,5] ITMO University, Saint Petersburg, 197101, Russian Federation

[1] viktor.zhuravlev2003@mail.ru, https://orcid.org/0009-0009-0788-9790
[2] arkhairullin@itmo.ru, https://orcid.org/0009-0007-0442-7764
[3] tsenred@yandex.com, https://orcid.org/0009-0000-3865-4446
[4] sitkina.alena2017@yandex.ru, https://orcid.org/0009-0002-6046-8943
[5] kylin98@list.ru✉, https://orcid.org/0000-0002-3952-6080

Abstract

Autoprompting is the process of automatically selecting optimized prompts for language models which is gaining popularity due to the rapid development of prompt engineering driven by extensive research in the field of Large Language Models. This paper presents DistillPrompt — a novel autoprompting method based on Large Language Models that employs a multi-stage integration of task-specific information into prompts using training data. DistillPrompt utilizes distillation, compression, and aggregation operations to explore the prompt space more thoroughly. The method was tested on different datasets for text classification and generation tasks using t-lite-instruct-2.1, gpt-3.5-turbo and gpt-4o-mini language models. The results demonstrate a significant average improvement in key metrics over existing methods in the field, establishing DistillPrompt as one of the most effective non-gradient approaches in autoprompting.

Keywords

LLM, autoprompting, prompt distillation, prompting, prompt engineering

For citation: Zhuravlev V.N., Khairullin A.R., Dyagin E.A., Sitkina A.N., Kulin N.I. Automatic prompt optimization with prompt distillation. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2026, vol. 26, no. 4, pp. 844–850. doi: 10.17586/2226-1494-2026-26-4-844-850

УДК 004.89

Автоматическая оптимизация промпта с помощью метода дистилляции

Виктор Николаевич Журавлев[1], Артур Рустамович Хайруллин[2], Эрнест Александрович Дягин[3], Алена Николаевна Ситкина[4], Никита Игоревич Кулин[5]✉

[1,2,3,4,5] Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация

[1] viktor.zhuravlev2003@mail.ru, https://orcid.org/0009-0009-0788-9790
[2] arkhairullin@itmo.ru, https://orcid.org/0009-0007-0442-7764
[3] tsenred@yandex.com, https://orcid.org/0009-0000-3865-4446
[4] sitkina.alena2017@yandex.ru, https://orcid.org/0009-0002-6046-8943
[5] kylin98@list.ru✉, https://orcid.org/0000-0002-3952-6080

Аннотация

Автопромптинг — процесс автоматического подбора оптимизированных промптов к языковым моделям, который набирает популярность в связи с быстрым развитием промпт инжиниринга, обусловленного многочисленными исследованиями в области больших языковых моделей. В работе представлен DistillPrompt — новый метод автопромптинга на основе больших языковых моделей, использующий многоступенчатое внедрение информации о задаче в промпт с применением тренировочной выборки. Разработанный метод использует операции дистилляции, сжатия, агрегации для более глубокого исследования пространства промптов. DistillPrompt протестирован на различных наборах данных по задачам классификации и генерации текста с применением языковых моделей t-lite-instruct-2.1, gpt-3.5-turbo and gpt-4o-mini. Предложенный метод показывает в среднем

значительный прирост метрик относительно текущих методов в данной области. Представленный подход является одним из самых эффективных вариантов автопромптинга на основе эволюционных алгоритмов.

Ключевые слова
БЯМ, автоматический промптинг, дистилляция промпта, промптинг, промпт-инжиниринг

Ссылка для цитирования: Журавлев В.Н., Хайруллин А.Р., Дягин Э.А., Ситкина А.Н., Кулин Н.И. Автоматическая оптимизация промпта с помощью метода дистилляции // Научно-технический вестник информационных технологий, механики и оптики. 2026. Т. 26, № 4. С. 844–850 (на англ. яз.). doi: 10.17586/2226-1494-2026-26-4-844-850

Introduction

In recent years, significant progress has been made in the field of text processing and generation using Artificial Intelligence (AI) — particularly Large Language Models (LLMs) [1–2]. Improving model output quality without modifying its weights falls under the domain of prompt engineering. This field employs various prompting techniques, including Few-shot [3], Chain-of-Thought [4], Directional Stimulus [5], among others. Research indicates that, depending on the task, these techniques can either enhance or significantly degrade model performance. For instance, applying Chain-of-Thought in tasks where reasoning may lead to incorrect answers can result in accuracy drops of tens of percentage points [6]. Similarly, with Few-shot prompting, it was found that for the Deepseek-R1 model, adding examples to the prompt "consistently degrades its performance" compared to a zero-shot task description [7].

To address this issue, autoprompting methods have emerged algorithms that leverage both the model itself and various heuristics to automatically improve prompt quality. Studies have shown that prompts generated by these methods often outperform those crafted by humans, even when designed by domain experts [8]. It is worth noting that the number of prompting techniques continues to grow each year, making manual prompt engineering increasingly complex and time-consuming. Thus, the challenge of autoprompting remains highly relevant.

This paper presents a novel and more effective non-gradient-based autoprompting approach. The core idea of our method is a complex prompt distillation which includes: generating diverse prompt candidates, injecting task-relevant examples from a subset of the training data into the prompt, aggregating candidates into a final optimized prompt, and iteratively refining candidates from the final prompt. The proposed approach was evaluated on different datasets and demonstrated superior performance compared to existing non-gradient autoprompting methods.

Non-gradient autoprompting methods

In recent years, there has been significant growth in research on LLMs and their applications across various domains. Given that prompting is an integral part of working with these models, numerous studies have explored autoprompting methods.

The first such approach was introduced in [9], which relied on fine-tuning an LLM to predict trigger tokens via softmax. However, this method had several limitations as computational overhead when the LLM required retraining and gradient updates to predict tokens and lack of interpretability where the generated trigger tokens were not human-interpretable, making it difficult to logically justify their effectiveness, even though most LLMs are inherently black-box models.

Subsequently, non-gradient-based autoprompting algorithms emerged, eliminating the need for gradient updates while allowing the extraction of interpretable prompt patterns for manual refinement [10, 11]. These approaches leverage semantic parsers, specialized prompt templates, LLMs themselves as the "brain" of the autoprompting algorithm. However, prior autoprompting methods suffer from several key drawbacks: insufficient prompt manipulation — limited transformations applied to the prompt structure, randomized instruction selection — arbitrary modification of prompt components without systematic optimization, narrow task applicability — restricted effectiveness across diverse Natural Language Processing (NLP) tasks. This paper addresses these limitations by proposing a more structured and generalizable non-gradient approach.

DistillPrompt

This paper presents an approach that addresses the limitations outlined in the previous section — DistillPrompt, illustrated in Figure. The method is based on prompt distillation and incorporates ideas from the

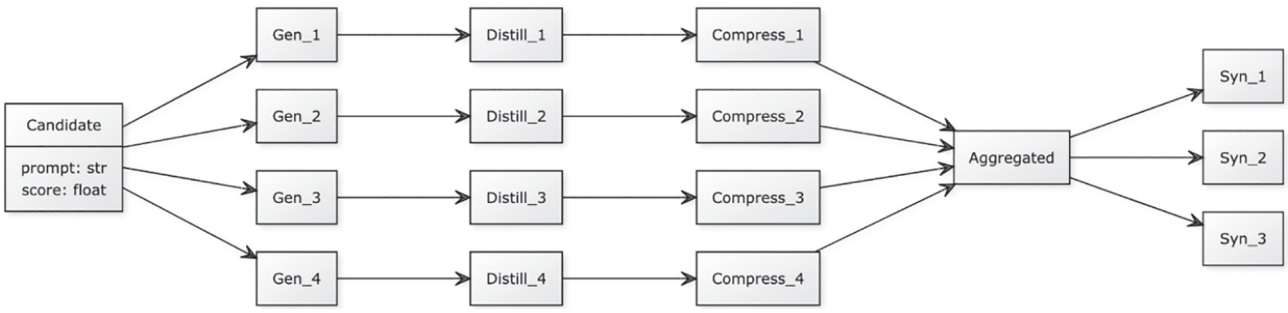

Figure. Workflow of DistillPrompt

Tree-of-Thoughts prompting technique [12, 13]. Prompt distillation refers to manipulating instructions through text compression, reformulating task descriptions, and incorporating usage examples. DistillPrompt is an iterative approach where each iteration consists of five sequential stages.

Stage 1. The initial best candidate is the provided prompt, and in each subsequent epoch, the best candidate from the previous iteration is used, where the best candidate is defined as the one with the highest target metric score on the training set.

At the start of an epoch, variations of the initial prompt are generated. This stage involves creating diverse modifications of the best candidate to explore the task from different perspectives. The purpose is to "explore" the space of potentially effective prompts and avoid local optima. As a result, N — new prompt candidates are produced. The number of candidates is a hyperparameter of the algorithm, balancing the trade-off between the number of LLM calls and the coverage of the prompt space for the given task. Generation is performed via queries to the LLM with a temperature of 0.7 (dimensionless) to enhance creativity while minimizing the risk of generating uninterpretable prompts.

Stage 2. The next stage is example embedding. While the previous step yielded four new prompt candidates, they explored the prompt space "blindly". To guide them toward the target task while preserving their unique formulations, we propose embedding examples from the training set. Initially, we tested direct example insertion (as in one-shot and few-shot techniques), but this proved less effective than using the LLM to analyze examples and extract their underlying task-solving principles, which better captures task-relevant information. For each prompt candidate, K examples are independently and randomly selected from the training set to guide the LLM in refining the prompt. However, there is a risk of the LLM "overfitting" to the examples — focusing on their specific labels and questions rather than deriving generalizable insights.

Stage 3. To mitigate this, the next stage involves instruction compression, where the LLM condenses the prompts from the previous step into a few sentences retaining the core ideas introduced by the examples and the overarching task objective. This step helps generalize the prompts while preserving the insights gleaned from the examples.

Stage 4. Next, candidate aggregation is performed. Since examples were selected independently and randomly for each candidate, the extracted insights vary. Thus, the natural progression is to merge the compressed candidates into a single distilled prompt encompassing the collective ideas.

Stage 5. The final stage generates new candidates from the distilled prompt by creating variations as in Stage 1. The resulting candidates are evaluated on tasks, and the top-performing candidate becomes the new initial prompt for the next epoch until the epoch limit is reached. Since the process explores the prompt space, candidates may outperform or underperform those from previous epochs; thus, the method requires multiple iterations. The algorithm output is the best prompt from the final epoch.

Experimental setup

In order to validate the effectiveness of DistillPrompt, we designed the following experimental setup: each evaluated method was tested across multiple datasets using the following LLMs: t-lite-instruct-2.1, gpt-3.5-turbo, gpt-4o-mini. The generation parameter config was as follows: temperature — 0.7 (dimensionless), maximum new tokens — 4,000.

For a robust evaluation framework, we analyzed multiple autoprompting studies [11–13] to compile diverse datasets. The resulting benchmark encompasses different tasks from classification to generation tasks. The complete dataset list includes:

— SST-2 — Binary sentiment binary classification task on movie review sentences from the Stanford Sentiment Treebank;
— MNLI — Multi-Genre Natural Language Inference task: given a premise and hypothesis, classify their relationship as entailment, contradiction, or neutral;
— TREC — Question classification dataset where questions are categorized into broad types (e.g., location, person, numeric value);
— MR — Movie Reviews sentiment binary classification dataset with one sentence per review;
— BBH (BIG-Bench Hard) — A curated subset of challenging tasks from the BIG-Bench benchmark that are difficult for LLMs;
— GSM8K — Grade school math word problems requiring multi-step reasoning to solve;
— CommonGen — Constrained text generation task: create coherent sentences using a provided set of common concepts;
— SQuAD v2 — Reading comprehension dataset with questions posed on Wikipedia articles, includes unanswerable questions;
— XSum — Extreme summarization dataset: generate concise, one-sentence summaries from BBC news articles;
— MedAlpaca — Medical instruction-following dataset for training AI assistants on healthcare topics, built from medical resources.

In this benchmark, question-answering datasets involve multiple-choice responses, making them effectively multiclass classification tasks. The benchmark datasets can be broadly categorized into classification and generation tasks, each requiring specific evaluation metrics. For SST-2, MNLI, TREC, MR and BBH classification tasks macro F1-score was used, for GSM8K — Exact Match was used, and for the rest datasets — BertScore.

The baseline comparisons include prompting techniques (baseline prompt — original dataset-provided prompt, few-shot prompt — baseline prompt augmented with three training examples) and following autoprompting approaches — Grips, Protegi and CriSPO. This experimental design enables systematic comparison of DistillPrompt against both manual prompting techniques and state-of-the-art autoprompting methods across diverse NLP tasks.

Ablation study on DistillPrompt parameters

Main DistillPrompt hyperparameters are a number of examples (denoted as *K*) and a number of prompt candidates (denoted as *N*). *K* is needed on a candidate generation step *Gen_i* and creating final diverse variants *Syn_i*, mentioned in section earlier.

In order to establish the most effective parameters, we conduct the auxiliary experiment to measure effectiveness of DistillPrompt on different configurations using classification datasets. The results of this are presented in Table 1.

The experiment results show a quality improvement when increasing *N* from 3 to 5 and *K* from 1 to 3, and then from 3 to 5. Further increasing *N* and *K* to 7 does not lead to significant improvement and in some cases results in degradation of the metrics. The highest average result was achieved with the configuration $N = 5$ and $K = 5$. Based on these results, we used this configuration in the subsequent experiment.

Results

The experimental results across metrics and datasets are presented in Tables 2–3 for different LLMs, showing performance on classification and generation tasks respectively. Table results performed the average score of three independent tasks. For ease of comparison, Tables 2 and 3 use the same set of method columns. Protegi was not evaluated on generation tasks because the method is not designed for them; therefore, its entries in Table 3 are marked with dashes.

Discussion

The conducted experiments demonstrate that DistillPrompt effectively handles both classification and text generation tasks. Across all evaluated datasets, DistillPrompt either outperformed or matched the performance of existing non-gradient autoprompting methods.

For classification tasks, the average F1-score improved by 13.78 % comparing baseline prompt and by 1.21 % comparing the strongest of baselines — CriSPO. In text generation tasks, the average score increased by 7.59 % comparing baseline prompt and by 1.34 % comparing the strongest of baselines — CriSPO. Comparisons and improvements were calculated relative to the maximum average metrics achieved by existing solutions.

This work creates a scope for future research into distillation of prompts and other non-gradient autoprompting methods. The current DistillPrompt implementation could potentially be further refined for more targeted prompt optimization. Moreover, the concept of prompt distillation could be generalized and adapted to other non-gradient autoprompting methods, representing a promising direction for future studies.

Table 1. DistillPrompt effectiveness analysis with different parameters. Result values indicate an average of three independent runs

Model	*N*	*K*	Dataset				
			SST-2	MNLI	TREC	MR	BBH (cls)
gpt-3.5-turbo	3	1	0.923	0.814	0.787	0.885	0.617
	3	3	0.944	0.823	0.812	0.928	0.654
	3	5	0.947	0.827	0.826	0.932	0.663
	5	1	0.935	0.823	0.814	0.925	0.652
	5	3	0.954	0.830	0.824	0.941	0.670
	5	5	**0.958**	**0.832**	**0.826**	0.952	**0.675**
	5	7	0.956	0.830	0.825	0.953	0.673
	7	1	0.936	0.825	0.816	0.927	0.655
	7	3	0.953	0.831	0.821	0.942	0.671
	7	5	**0.958**	0.831	0.824	**0.953**	0.674
gpt-4o-mini	3	1	0.951	0.890	0.913	0.943	0.810
	3	3	0.961	0.832	0.916	0.928	0.818
	3	5	0.973	0.854	0.921	0.933	0.802
	5	1	0.974	0.913	0.927	0.955	0.825
	5	3	0.965	0.915	0.925	0.956	0.823
	5	5	0.980	**0.920**	**0.941**	**0.970**	**0.834**
	5	7	0.978	0.919	0.940	0.968	0.831
	7	1	0.975	0.913	0.929	0.956	0.826
	7	3	0.967	0.917	0.925	0.953	0.825
	7	5	**0.981**	0.914	0.939	**0.970**	0.832

Note: Bold values indicate results that outperform others. *K* is a number of examples; *N* is a number of prompt candidates.

Table 2. DistillPrompt comparative analysis between autoprompting methods on classification tasks

Model	Dataset	Method					
		Baseline prompt	Few shot (3)	Protegi	Grips	CriSPO	DistillPrompt
t-lite-instruct-2.1	SST-2	0.613	0.933	0.640	0.614	0.936	**0.949**
	MNLI	0.418	0.374	0.496	0.741	0.758	**0.761**
	TREC	0.728	0.734	**0.771**	0.743	0.755	0.763
	MR	0.862	0.603	0.636	0.918	0.930	**0.939**
	BBH (cls)	0.206	0.313	0.372	0.288	0.388	**0.405**
gpt-3.5-turbo	SST-2	0.902	0.910	0.890	0.878	0.954	**0.958**
	MNLI	0.723	0.750	0.783	0.810	0.823	**0.832**
	TREC	0.787	0.793	0.814	0.799	0.804	**0.826**
	MR	0.890	0.893	0.925	0.920	0.945	**0.952**
	BBH (cls)	0.610	0.643	0.652	0.624	0.663	**0.675**
gpt-4o-mini	SST-2	0.933	0.943	0.965	0.974	**0.983**	0.980
	MNLI	0.832	0.854	0.880	0.895	0.907	**0.920**
	TREC	0.916	0.921	0.932	0.925	0.938	**0.941**
	MR	0.928	0.933	0.960	0.956	0.941	**0.970**
	BBH (cls)	0.818	0.802	0.810	0.823	0.828	**0.834**

Note: Bold values indicate results that outperform others.

Table 3. DistillPrompt comparative analysis between autoprompting methods on generation tasks

Model	Dataset	Method					
		Baseline prompt	Few shot (3)	Protegi	Grips	CriSPO	DistillPrompt
t-lite-instruct-2.1	GSM8K	0.674	0.680	—	0.704	0.715	**0.718**
	BBH (gen)	0.725	0.730	—	0.752	0.748	**0.754**
	CommonGen	0.764	0.775	—	0.815	0.820	**0.823**
	SQUAD v2	0.826	0.834	—	0.850	0.849	**0.853**
	XSUM	0.680	0.625	—	0.710	0.724	**0.730**
	MedAlpaca	0.545	0.590	—	0.634	**0.725**	0.713
gpt-3.5-turbo	GSM8K	0.724	0.753	—	0.824	0.832	**0.853**
	BBH (gen)	0.802	0.808	—	0.810	0.821	**0.833**
	CommonGen	0.807	0.815	—	0.832	0.854	**0.911**
	SQUAD v2	0.909	0.904	—	0.918	0.910	**0.922**
	XSUM	0.802	0.815	—	0.821	0.839	**0.842**
	MedAlpaca	0.709	0.720	—	0.732	0.783	**0.801**
gpt-4o-mini	GSM8K	0.856	0.842	—	0.889	0.884	**0.901**
	BBH (gen)	0.838	0.841	—	0.865	0.870	**0.873**
	CommonGen	0.885	0.893	—	0.914	0.921	**0.935**
	SQUAD v2	0.891	0.905	—	0.905	0.918	**0.931**
	XSUM	0.821	0.825	—	0.835	0.842	**0.851**
	MedAlpaca	0.853	0.870	—	0.918	0.927	**0.939**

Note: Bold values indicate results that outperform others. Dashes indicate that Protegi was not evaluated on generation tasks because the method is not designed for them.

Conclusion

The proposed DistillPrompt algorithm, which employs distillation prompt technique for prompt optimization, was evaluated on classification and generation datasets covering various natural language processing domains. It demonstrated consistent improvements over existing non-gradient algorithm-based autoprompting methods. DistillPrompt proves to be a competitive solution, showing that exploring prompt distillation for autoprompting can yield significant benefits and advance current methods to new levels of performance.

References

1. Kadavath S., Conerly T., Askell A., Henighan T., Drain D., Perez E., et al. Language models (mostly) know what they know. *arXiv*, 2022. arXiv:2207.05221. doi: 10.48550/arXiv.2207.05221
2. Wei J., Bosma M., Zhao V.Y., Guu K., Yu A.W., Lester B., et al. Finetuned language models are zero-shot learners. *arXiv*, 2021. arXiv:2109.01652. doi: 10.48550/arXiv.2109.01652
3. Brown T.B., Mann B., Ryder N., Subbiah M., Kaplan J., Dhariwal P., Neelakantan A., et al. Language models are few-shot learners. *arXiv*, 2020. arXiv:2005.14165. doi: 10.48550/arXiv.2005.14165
4. Wei J., Wang X., Schuurmans D., Bosma M., Ichter B., Xia F., et al. Chain-of-thought prompting elicits reasoning in large language models. *Proc. of the 36th International Conference on Neural Information Processing Systems*, 2022, pp. 24824–24837.
5. Li Z., Peng B., He P., Galley M., Gao J., Yan X. Guiding large language models via directional stimulus prompting. *Proc. of the 37th International Conference on Neural Information Processing Systems*, 2023, pp. 62630–62656.
6. Liu R., Geng J., Wu A.J., Sucholutsky I., Lombrozo T., Griffiths T.L. Mind your step (by step): chain-of-thought can reduce performance on tasks where thinking makes humans worse. *Proc. of the 42nd International Conference on Machine Learning*, 2025, pp. 38489–38517.
7. Guo D., Yang D., Zhang H., Song J., Wang P., Zhu Q., et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 2025, vol. 645, no. 8081, pp. 633–638. doi: 10.1038/s41586-025-09422-z
8. Zhou Y., Muresanu A.I., Han Z., Paster K., Pitis S., Chan H., et al. Large language models are human-level prompt engineers. *Proc. of the 11th International Conference on Learning Representations*, 2023.
9. Shin T., Razeghi Y., Logan R.L., Wallace E., Singh S. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *Proc. of the Conference on Empirical Methods in Natural Language Processing Proceedings of the Conference (EMNLP-2020)*, 2020, pp. 4222–4235.
10. Prasad A., Hasa P., Zhou X., Bansal M. Grips: Gradient-free, edit-based instruction search for prompting large language models. *Proc. of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 2023, pp. 3845–3864. doi: 10.18653/v1/2023.eacl-main.277
11. Pryzant R., Iter D., Li J., Lee Y., Zhu C., Zeng M. Automatic prompt optimization with "gradient descent" and beam search. *Proc. of the Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 7957–7968. doi: 10.18653/v1/2023.emnlp-main.494
12. Li L., Zhang Y., Chen L. Prompt distillation for efficient llm-based recommendation. *Proc. of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 1348–1357. doi: 10.1145/3583780.3615017
13. Yao S., Yu D., Zhao J., Shafran I., Griffiths T., Cao Y., et al. Tree of thoughts: Deliberate problem solving with large language models. *Proc. of the 37th International Conference on Neural Information Processing Systems*, 2023, pp. 11809–11822.

Литература

1. Kadavath S., Conerly T., Askell A., Henighan T., Drain D., Perez E., et al. Language models (mostly) know what they know // arXiv. 2022. arXiv:2207.05221. doi: 10.48550/arXiv.2207.05221
2. Wei J., Bosma M., Zhao V.Y., Guu K., Yu A.W., Lester B., et al. Finetuned language models are zero-shot learners // arXiv. 2021. arXiv:2109.01652. doi: 10.48550/arXiv.2109.01652
3. Brown T.B., Mann B., Ryder N., Subbiah M., Kaplan J., Dhariwal P., Neelakantan A., et al. Language models are few-shot learners // arXiv. 2020. arXiv:2005.14165. doi: 10.48550/arXiv.2005.14165
4. Wei J., Wang X., Schuurmans D., Bosma M., Ichter B., Xia F., et al. Chain-of-thought prompting elicits reasoning in large language models // Proc. of the 36th International Conference on Neural Information Processing Systems. 2022. P. 24824–24837.
5. Li Z., Peng B., He P., Galley M., Gao J., Yan X. Guiding large language models via directional stimulus prompting // Proc. of the 37th International Conference on Neural Information Processing Systems. 2023. P. 62630–62656.
6. Liu R., Geng J., Wu A.J., Sucholutsky I., Lombrozo T., Griffiths T.L. Mind your step (by step): chain-of-thought can reduce performance on tasks where thinking makes humans worse // Proc. of the 42nd International Conference on Machine Learning. 2025. P. 38489–38517.
7. Guo D., Yang D., Zhang H., Song J., Wang P., Zhu Q., et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning // Nature. 2025. V. 645. N 8081. P. 633–638. doi: 10.1038/s41586-025-09422-z
8. Zhou Y., Muresanu A.I., Han Z., Paster K., Pitis S., Chan H., et al. Large language models are human-level prompt engineers // Proc. of the 11th International Conference on Learning Representations. 2023.
9. Shin T., Razeghi Y., Logan R.L., Wallace E., Singh S. Autoprompt: Eliciting knowledge from language models with automatically generated prompts // Proc. of the Conference on Empirical Methods in Natural Language Processing Proceedings of the Conference (EMNLP-2020). 2020. P. 4222–4235.
10. Prasad A., Hasa P., Zhou X., Bansal M. Grips: Gradient-free, edit-based instruction search for prompting large language models // Proc. of the 17th Conference of the European Chapter of the Association for Computational Linguistics. 2023. P. 3845–3864. doi: 10.18653/v1/2023.eacl-main.277
11. Pryzant R., Iter D., Li J., Lee Y., Zhu C., Zeng M. Automatic prompt optimization with "gradient descent" and beam search // Proc. of the Conference on Empirical Methods in Natural Language Processing. 2023. P. 7957–7968. doi: 10.18653/v1/2023.emnlp-main.494
12. Li L., Zhang Y., Chen L. Prompt distillation for efficient llm-based recommendation // Proc. of the 32nd ACM International Conference on Information and Knowledge Management. 2023. P. 1348–1357. doi: 10.1145/3583780.3615017
13. Yao S., Yu D., Zhao J., Shafran I., Griffiths T., Cao Y., et al. Tree of thoughts: Deliberate problem solving with large language models // Proc. of the 37th International Conference on Neural Information Processing Systems. 2023. P. 11809–11822.

Authors

Viktor N. Zhuravlev — Student, ITMO University, Saint Petersburg, 197101, Russian Federation, https://orcid.org/0009-0009-0788-9790, viktor.zhuravlev2003@mail.ru

Artur R. Khairullin — Student, ITMO University, Saint Petersburg, 197101, Russian Federation, https://orcid.org/0009-0007-0442-7764, arkhairullin@itmo.ru

Авторы

Журавлев Виктор Николаевич — студент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, https://orcid.org/0009-0009-0788-9790, viktor.zhuravlev2003@mail.ru

Хайруллин Артур Рустамович — студент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, https://orcid.org/0009-0007-0442-7764, arkhairullin@itmo.ru

Ernest A. Dyagin — Student, ITMO University, Saint Petersburg, 197101, Russian Federation, https://orcid.org/0009-0000-3865-4446, tsenred@yandex.com
Alyona N. Sitkina — Student, ITMO University, Saint Petersburg, 197101, Russian Federation, https://orcid.org/0009-0002-6046-8943, sitkina.alena2017@yandex.ru
Nikita I. Kulin — PhD Student, ITMO University, Saint Petersburg, 197101, Russian Federation, sc 57222386134, https://orcid.org/0000-0002-3952-6080, kylin98@list.ru

Дягин Эрнест Александрович — студент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, https://orcid.org/0009-0000-3865-4446, tsenred@yandex.com
Ситкина Алена Николаевна — студент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, https://orcid.org/0009-0002-6046-8943, sitkina.alena2017@yandex.ru
Кулин Никита Игоревич — аспирант, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, sc 57222386134, https://orcid.org/0000-0002-3952-6080, kylin98@list.ru

Received 12.07.2025
Approved after reviewing 07.02.2026
Accepted 22.07.2026

Статья поступила в редакцию 12.07.2025
Одобрена после рецензирования 07.02.2026
Принята к печати 22.07.2026