УДК *004.89*

В. И. Коробцов [1], И.В. Овсянников [1], Д.И. Сачков [1]

[1] Иркутский государственный университет путей сообщения, г. Иркутск, Российская Федерация

АВТОМАТИЧЕСКАЯ ГЕНЕРАЦИЯ НАДЕЖНОГО ПРОГРАММНОГО КОДА С ПОМОЩЬЮ ГЕНЕРАТИВНЫХ ПРЕДОБУЧЕННЫХ ТРАНСФОРМЕРОВ (GPT)

Аннотация. В данном исследовании авторы фокусируются на оценке способности различных моделей генеративных предобученных трансформеров (GPT) автоматически генерировать программный код для решения задач алгоритмического характера с различным уровнем сложности, выбранных с платформы Leetcode. Рассматриваемые модели включают: GPT-3.5 и GPT-4 от OpenAI, Yandex-GPT от российского разработчика Yandex, GigaChat от SBER, Gemini от Google, Copilot от Microsoft, Mistral AI от французкой компании Mistral, Claude-3 от Anthropic AI и модель с открытым исходным кодом Llama-70b. GPT-3.5 рассматривается как устаревшая модель, тогда как GPT-4 представляет собой последнюю разработку OpenAI. Yandex-GPT и GigaChat выделяются как продукты общего назначения с упором на корпоративное использование, разработанные в России. Модель Copilot от Microsoft спроектирована специально для поддержки разработчиков в написании программного кода в интегрированных средах разработки.

Ключевые слова: *Искусственный интеллект, генеративные предобученные трансформеры, GPT-3.5, GPT-4, Yandex-GPT, GigaChat, Gemini, Copilot Claude-3, open source, автоматическая генерация кода, Leetcode*

V. I. Korobtsov [1], I.V. Ovsyannikov [1], D.I. Sachkov [1]

[1] Irkutsk State Transport University, Irkutsk, Russian Federation

AUTOMATIC GENERATION OF RELIABLE PROGRAM CODE USING GENERATIVE PRE-TRAINED TRANSFORMERS (GPT)

Abstract. In this article, the authors focus on assessing the ability of various Generative Pretrained Transformers (GPT) models to automatically generate code for solving algorithmic problems of varying complexity selected from the Leetcode platform. Models under consideration include: GPT-3.5 and GPT-4 from OpenAI, Yandex-GPT from Russian developer Yandex, GigaChat from SBER, Gemini from Google, Copilot from Microsoft, Mistral AI from the French company Mistral, Claude-3 from Anthropic AI and a model with open-source Llama-70b. GPT-3.5 is considered a legacy model, while GPT-4 is the latest development from OpenAI. Yandex-GPT and GigaChat stand out as general-purpose, enterprise-focused products developed in Russia. Microsoft's Copilot model is designed specifically to help developers write code in integrated development environments.

Keywords: *Artificial intelligence, generative pre-trained transformers, GPT-3.5, GPT-4, Yandex-GPT, GigaChat, Gemini, Copilot Claude-3, open source, automatic code generation, Leetcode*

Введение. В последнем десятилетии, в свете стремительного развития технологий искусственного интеллекта (ИИ) и увеличения требований к профессиональным навыкам разработчиков информационных систем, вопрос автоматической генерации программного кода обрел актуальность. На передний план вышли генеративные предобученные трансформеры (GPT) [1], представляющие собой модели, способные обрабатывать запросы на естественном языке [2]. Такие системы, насыщенные обширными базами как размеченных, так и неразмеченных данных, обладают способностью генерировать функционально полноценный и качественный программный код [3]. В исследовании [4] анализируется влияние автоматической генерации кода на элементы критической инфраструктуры, при этом выделяются риски и разрабатываются методы предотвращения возможных катастрофических сбоев. Авторы предлагают инновационный подход к проектированию программного обеспечения, который базируется на корректном формировании запросов (Prompt) к предобученным моделям для инициирования генерации высококачественного кода. В работе [5] авторы проверяют возможность генерации нетривиальных параллельных NPDP алгоритмов, с помощью модели GPT 3 от компании OpenAI. В исследовании [6] авторами отмечается, что разные модели GPT в целом справляются с национальным медицинским

экзаменом, проводимым в Перу. В работах [7-9] освещается морально-этический аспект применения предобученных моделей, исследуется их воздействие на человеческий быт и профессиональную деятельность. Обсуждается, как применение данных технологий влияет на качество и эффективность обучения, а также на способность к адаптации и освоению новых знаний. Особое внимание уделяется потенциальному риску коррупции в образовательном процессе, связанному с использованием искусственного интеллекта для автоматизации учебных задач. В данной же работе приводится обзор существующих больших языковых моделей (LLM), способных обрабатывать запросы на естественном языке. Далее проводится сравнительный анализ, в котором сравнивается корректность, надежность и эффективность сгенерированного программного кода.

Перечень доступных сегодня моделей. В таблице 1 приведен список самых популярных (в том числе российских) GPT, доступных сегодня.

Таблица 1.

Исследуемые модели GPT

№	Название GPT	Примечание
1	GPT-3.5	Устаревшая модель от OpenAI
2	GPT-4	Последняя версия модели от OpenAI
3	Yandex-GPT	Модель GPT общего назначения от российского разработчика Yandex с упором на корпоративное использование
4	GigaChat	Модель GPT общего назначения от российского разработчика SBER
5	Gemini	Модель от компании Google
6	Copilot	Помощник для написания кода от Microsoft
7	Mistral AI	Модель от французской компании Mistral
8	Claude-3	Модель от компании Anthropic AI
9	Llama-70b	Open-source модель

Модели GPT-3.5 и GPT-4, разработанные американской компанией OpenAI, занимают важное место в исследовании современных возможностей искусственного интеллекта. Из работы [10] следует, что GPT-4 значительно превосходит своего предшественника, GPT-3, обладая значительно увеличенным числом внутренних параметров. Согласно представленным данным (таблица 2), объем параметров модели GPT-4 превышает аналогичный показатель GPT-3 в 2.8 раза и на 145 миллиардов больше, чем у GPT-3.5. Эта колоссальная мощность позволяет GPT-4 выходить за рамки генерации текстовых ответов, давая возможность создавать изображения и звук, а также обрабатывать запросы на основе различных текстовых документов, например, PDF.

Таблица 2.

Количество параметров у различных GPT от OpenAI

GPT-1	GPT-2	GPT-3	GPT-3.5	GPT-4
117 млн.	1.5 млрд.	175 млрд.	355 млрд.	500 млрд.

В исследовании [11] сравниваются способности GPT-4 от OpenAI и Gemini от Google в понимании общенаучных вопросов. Обе модели демонстрируют высокую компетентность, в некоторых случаях превосходя человеческий уровень понимания. Различия между моделями проявляются в стиле их ответов: GPT-4 характеризуется краткостью и точностью в представлении информации, в то время как Gemini выделяется более подробными и исчерпывающими ответами. Это различие может быть связано с уникальными архитектурными и обучающими особенностями каждой из моделей. Модель GPT-4, с ее огромным числом параметров, оптимизируется для сжатого и точного представления информации, тогда как Gemini, будучи натренированной на более широком спектре мультимодальных данных, вероятно склонна к более глубокому анализу и подробному изложению материала.

Copilot, разработанный компанией Microsoft, позиционируется как инструмент искусственного интеллекта для автодополнения, он предлагает разработчикам варианты для завершения строк кода. Эта модель интегрируется с различными интегрированными средами разработки (IDE), например, Visual Studio Code. Исследования [12-13] указывают на двойственность Copilot в его использовании: с одной стороны, он может существенно ускорить процесс написания кода, предлагая решения на основе контекста; с другой стороны, не всегда предложенные решения оказываются корректными. Разработчику нужно тщательно оценивать предлагаемые варианты и принимать осознанное решение об их использовании. Кроме того, в отличие от более гибких моделей GPT, Copilot обладает ограничениями в интерактивности. Например, в нем отсутствует поддержка уточнений в свободной форме на естественном языке ("prompt refinement").

В исследовании [14] рассматриваются ключевые различия между российской моделью YandexGPT и ChatGPT (GPT-3) от OpenAI. YandexGPT преимущественно обучалась на данных на русском языке, что делает её особенно эффективной для русскоязычных пользователей. Эта модель тесно интегрирована в экосистему продуктов Yandex и взаимодействует с голосовым помощником «Алиса», предоставляя широкий спектр услуг и функций, адаптированных под нужды пользователей Yandex. Однако в работе показано, что чаще всего, GPT-3 демонстрирует более высокую точность в обработке запросов.

Генерация программного кода представленными моделями. Для генерации программного кода был использован набор задач из таблицы 3, взятый с известной интернет-платформы Leetcode [15], специализирующейся на алгоритмических задачах для разработчиков. Задачи классифицированы по уровню сложности на три категории: "Простые" (Easy), "Средние" (Medium) и "Сложные" (Hard). Для каждой задачи указывается процент её решений пользователями платформы, что демонстрирует общую успешность решения задачи среди участников сообщества. Корректность предложенных решений автоматически верифицируется платформой посредством заранее определенного набора тестов. Сами задания подробно описаны на сайте Leetcode, доступ к полным описаниям задач открыт для всех желающих. В рамках данного исследования, корректным будет считаться решение, сгенерированное моделью GPT, в виде работающего программного кода на языке программирования Python, который компилируется и проходит верификацию платформы LeetCode.

Таблица 3.

Практические задачи для решения

Название задачи	Сложность, процент решения
Two Sum	Easy, 52.2%
Roman to Integer	Easy, 61.0%
Search Insert Position	Easy, 45.7%
Add Two Numbers	Medium, 42.6%
Valid Number	Hard, 19.7%

Введем матрицу оценок каждой модели GPT A=$\left|\left|a_{ij}\right|\right|, i = \underline{1,9}, j = \underline{1,4}$, где i – номер оцениваемой модели, согласно таблице 1, j – номер решаемой задачи, согласно таблице 3, по правилу:

$$a_{ij}=\begin{cases} 2, & \text{если } i-\text{ая модель решила } j-\text{ую задачу с 1 раза} \\ 1, & \text{если } i-\text{ая модель решила } j-\text{ую задачу с доп. пояснениями} \\ 0, & \text{если } i-\text{ая модель не справилась с решением } j-\text{ой задачи} \end{cases}$$

Дополнительно, для понимания процесса генерации, рассмотрим программный код, сгенерированный моделями GPT-3, GPT-4, Gemini, YandexGPT, GigaChat для задачи «Two Sum». Условие задачи состоит в том, чтобы в данном массиве чисел найти такие два вхождения элементов, чтобы их сумма была равна данному числу N и вернуть их в виде списка. Например, пусть исходные данные заданы следующим образом:

$$nums = [2, 7, 11, 15];\ N = 9 \ (1)$$

Правильным решением будет список [0, 1] т.к. $nums[0] + nums[1] = 9$

Результат генерации моделью GPT-3.5

```
class Solution(object):
    def twoSum(self, nums, target):
        num_indices = {}
        for i, num in enumerate(nums):
            complement = target - num
            if complement in num_indices:
                return [num_indices[complement], i]
            num_indices[num] = i
```

Представленный программный код от модели GPT-3.5 прошел верификацию платформы. Код не снабжен никакими дополнительными комментариями.

Результат генерации моделью GPT-4

```
class Solution:
    def twoSum(self, nums, List[int], target: int) -> List[int]:
        num_to_index = {}
        for index, num in enumerate(nums):
            complement = target - num
            if complement in num_to_index:
                return [num_to_index[complement], index]
            num_to_index[num] = index
```

Код, представленный моделью GPT-4, также без проблем прошел верификацию. Он отличается от кода, сгенерированного моделью GPT-3 и является более технологичным, однако, это не повлияло на корректность решения.

Результат генерации моделью Gemini

```
class Solution(object):
    def twoSum(self, nums, target):
        seen = {}  # Dictionary to store previously seen numbers and their indices
        for i, num in enumerate(nums):
            complement = target - num  # Calculate the complement needed to reach the target
            if complement in seen:  # If the complement has been seen before, return the indices
                return [seen[complement], i]
            seen[num] = i  # Store the current number and its index in the dictionary for future
lookup
        return []  # No solution found
```

Код, представленный моделью Gemini, прошел верификацию платформы. Он также дополнительно снабжен поясняющими комментариями, в отличии от кода, представленного моделями GPT-3.5 и GPT-4.

Результат генерации моделью YandexGPT

```
class Solution(object):
    def twoSum(self, nums, target):
        lookup = {}
        for i, num in enumerate(nums):
            complement = target - num
            if complement in lookup:
                return [lookup[complement], i]
            lookup[num] = i
```

Модель YandexGPT также справилась с решением задачи с 1 раза, код не снабжен дополнительными комментариями.

Результат генерации моделью GigaChat

В ситуации с моделью GigaChat произошла неожиданность: несмотря на запрос, требующий код на Python, модель сгенерировала код на Java. Это указывает на потенциальное несоответствие в понимании контекста запроса или же на отсутствие гибкости модели в выборе языка программирования. Кроме того, в коде присутствуют три тестовых случая в методе main, что не соответствует требованиям задачи. Решение должно быть оформлено в отдельном методе twoSum без лишних элементов в классе. Из-за этого несоответствия, код не может быть верифицирован платформой, однако, после удаления метода main, код прошел верификацию платформы.

В таблице 4 представлены результаты проведенного исследования.

Таблица 4.

Матрица А оценок исследуемых моделей

Модель / Задача	Two Sum	Roman to Integer	Search Insert Position	Add Two Numbers	Valid Number	Кол-во баллов
GPT-3.5	2	2	2	1	1	8
GPT-4	2	2	2	2	2	9
Gemini	2	1	2	2	2	9
Yandex-GPT	2	0	2	0	2	6
GigaChat	1	0	2	1	1	5
Copilot	2	2	2	1	2	9
Mistral	2	2	2	2	1	9
Claude-3	2	2	2	2	2	10
Llama-70b	1	1	2	0	1	5

Согласно проведенному исследованию, в настоящее время, самой эффективной моделью GPT для генерации программного кода стоит признать модель Claude-3. Самыми неэффективными модели: GigaChat, Llama-70b. С осторожностью стоит относится ко всем остальным представленным в исследовании моделям.

Заключение. В рамках настоящего исследования представлены данные, которые могут быть интегрированы в процесс выбора адекватной языковой модели разработчиками программного обеспечения с целью генерации кода. В контексте интенсивного прогресса в области больших языковых моделей и искусственного интеллекта в целом, авторами настоятельно рекомендуется осуществлять всесторонний аналитический подход при

определении наиболее подходящей модели для применения в проектах с повышенными требованиями к надежности и функциональности программного кода.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. X. Han, Z. Zhang, N. Ding, Y. Gu, X. Liu, Y. Huoet al., "Pre-trained models: past, present and future", AI Open, vol. 2, p. 225-250, 2021. https://doi.org/10.1016/j.aiopen.2021.08.002
2. D. Otter, J. Medina, & J. Kalita, "A survey of the usages of deep learning for natural language processing", IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 2, p. 604-624, 2021. https://doi.org/10.1109/tnnls.2020.2979670
3. B. Idrisov and T. Schlippe, "Program code generation with generative ais", Algorithms, vol. 17, no. 2, p. 62, 2024. https://doi.org/10.3390/a17020062
4. M. Liu, J. Wang, T. Lin, Q. Ma, Z. Fang, & Y. Wu, "An empirical study of the code generation of safety-critical software using llms", Applied Sciences, vol. 14, no. 3, p. 1046, 2024. https://doi.org/10.3390/app14031046
5. M. Palkowski and M. Gruzewski, "Gpt-driven source-to-source transformation for generating compilable parallel cuda code for nussinov's algorithm", Electronics, vol. 13, no. 3, p. 488, 2024. https://doi.org/10.3390/electronics13030488
6. B. Torres-Zegarra, W. Rios-Garcia, A. Ñaña-Cordova, K. Arteaga-Cisneros, X. Chalco, M. Ordoñezet al., "Performance of chatgpt, bard, claude, and bing on the peruvian national licensing medical examination: a cross-sectional study", Journal of Educational Evaluation for Health Professions, vol. 20, p. 30, 2023. https://doi.org/10.3352/jeehp.2023.20.30
7. D. Cotton, P. Cotton, & J. Shipway, "Chatting and cheating: ensuring academic integrity in the era of chatgpt", Innovations in Education and Teaching International, vol. 61, no. 2, p. 228-239, 2023. https://doi.org/10.1080/14703297.2023.2190148
8. C. Awad, "The impact of gb chat on education - a comparative theoretical study", International Journal of Computers and Informatics, vol. 2, no. 6, p. 89-123, 2023. https://doi.org/10.59992/ijci.2023.v2n6p5
9. K. Kim, "Study on artificial intelligence(ai) and chat gpt, corruption", The Korea Association for Corruption Studies, vol. 28, no. 2, p. 85-105, 2023. https://doi.org/10.52663/kcsr.2023.28.2.85
10. M. Rahaman, M. Ahsan, N. Anjum, H. Terano, & M. Rahman, "From chatgpt-3 to gpt-4: a significant advancement in ai-driven nlp tools", Journal of Engineering and Emerging Technologies, vol. 1, no. 1, p. 50-60, 2023. https://doi.org/10.52631/jeet.v1i1.188
11. M. Nyaaba, "Comparing human and ai's (gpt-4 and gemini) understanding of the nature of science", SSRN Electronic Journal, 2023. https://doi.org/10.2139/ssrn.4661602
12. B. Zhang, P. Liang, X. Zhou, A. Ahmad, & M. Waseem, "Demystifying practices, challenges and expected features of using github copilot", International Journal of Software Engineering and Knowledge Engineering, vol. 33, no. 11n12, p. 1653-1672, 2023. https://doi.org/10.1142/s0218194023410048
13. Ziegler, E. Kalliamvakou, X. Li, A. Rice, D. Rifkin, S. Simisteret al., "Measuring github copilot's impact on productivity", Communications of the ACM, vol. 67, no. 3, p. 54-63, 2024. https://doi.org/10.1145/3633453
14. Солдатенкова, Ю. А. YandexGPT и ChatGPT: характеристика, сравнение и основные отличия нейросетей / Ю. А. Солдатенкова, А. В. Свищев // Моя профессиональная карьера. – 2023. – Т. 3, № 55. – С. 277-284.
15. LeetCode - The World's Leading Online Programming Learning Platform. - URL: https://leetcode.com/ (дата обращения: 01.04.2024).

REFERENCES

1. X. Han, Z. Zhang, N. Ding, Y. Gu, X. Liu, Y. Huoet al., "Pre-trained models: past, present and future", AI Open, vol. 2, p. 225-250, 2021. https://doi.org/10.1016/j.aiopen.2021.08.002

2. D. Otter, J. Medina, & J. Kalita, "A survey of the usages of deep learning for natural language processing", IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 2, p. 604-624, 2021. https://doi.org/10.1109/tnnls.2020.2979670
3. B. Idrisov and T. Schlippe, "Program code generation with generative ais", Algorithms, vol. 17, no. 2, p. 62, 2024. https://doi.org/10.3390/a17020062
4. M. Liu, J. Wang, T. Lin, Q. Ma, Z. Fang, & Y. Wu, "An empirical study of the code generation of safety-critical software using llms", Applied Sciences, vol. 14, no. 3, p. 1046, 2024. https://doi.org/10.3390/app14031046
5. M. Palkowski and M. Gruzewski, "Gpt-driven source-to-source transformation for generating compilable parallel cuda code for nussinov's algorithm", Electronics, vol. 13, no. 3, p. 488, 2024. https://doi.org/10.3390/electronics13030488
6. B. Torres-Zegarra, W. Rios-Garcia, A. Ñaña-Cordova, K. Arteaga-Cisneros, X. Chalco, M. Ordoñezet al., "Performance of chatgpt, bard, claude, and bing on the peruvian national licensing medical examination: a cross-sectional study", Journal of Educational Evaluation for Health Professions, vol. 20, p. 30, 2023. https://doi.org/10.3352/jeehp.2023.20.30
7. D. Cotton, P. Cotton, & J. Shipway, "Chatting and cheating: ensuring academic integrity in the era of chatgpt", Innovations in Education and Teaching International, vol. 61, no. 2, p. 228-239, 2023. https://doi.org/10.1080/14703297.2023.2190148
8. C. Awad, "The impact of gb chat on education - a comparative theoretical study", International Journal of Computers and Informatics, vol. 2, no. 6, p. 89-123, 2023. https://doi.org/10.59992/ijci.2023.v2n6p5
9. K. Kim, "Study on artificial intelligence(ai) and chat gpt, corruption", The Korea Association for Corruption Studies, vol. 28, no. 2, p. 85-105, 2023. https://doi.org/10.52663/kcsr.2023.28.2.85
10. M. Rahaman, M. Ahsan, N. Anjum, H. Terano, & M. Rahman, "From chatgpt-3 to gpt-4: a significant advancement in ai-driven nlp tools", Journal of Engineering and Emerging Technologies, vol. 1, no. 1, p. 50-60, 2023. https://doi.org/10.52631/jeet.v1i1.188
11. M. Nyaaba, "Comparing human and ai's (gpt-4 and gemini) understanding of the nature of science", SSRN Electronic Journal, 2023. https://doi.org/10.2139/ssrn.4661602
12. B. Zhang, P. Liang, X. Zhou, A. Ahmad, & M. Waseem, "Demystifying practices, challenges and expected features of using github copilot", International Journal of Software Engineering and Knowledge Engineering, vol. 33, no. 11n12, p. 1653-1672, 2023. https://doi.org/10.1142/s0218194023410048
13. Ziegler, E. Kalliamvakou, X. Li, A. Rice, D. Rifkin, S. Simisteret al., "Measuring github copilot's impact on productivity", Communications of the ACM, vol. 67, no. 3, p. 54-63, 2024. https://doi.org/10.1145/3633453
14. Soldatenkova, YU. A. YandexGPT i ChatGPT: harakteristika, sravnenie i osnovnye otlichiya nejrosetej [YandexGPT and ChatGPT: characteristics, comparison and main differences between neural networks], YU. A. Soldatenkova, A. V. Svishchev, Moya professional'naya kar'era. [My professional career]. 2023, I. 3, № 55, pp. 277-284. (In Russian)
15. LeetCode - The World's Leading Online Programming Learning Platform. - URL: https://leetcode.com/ (access date: 04/01/2024).

Информация об авторах

Коробцов Владислав Игоревич - студент 4 курса кафедры «Информационные технологии и защита информации», направления подготовки «Программная инженерия», Иркутский государственный университет путей сообщения

Овсянников Иван Владимирович – студент 4 курса кафедры «Информационные технологии и защита информации», направления подготовки «Программная инженерия», Иркутский государственный университет путей сообщения

Сачков Дмитрий Иванович – к. э. н., проректор по цифровым технологиям, Иркутский государственный университет путей сообщения, г. Иркутск, e-mail: sachkov_di@irgups.ru

Information about the authors

Korobtsov V.I. - 4th year student of the Department of Information Technologies and Information Security, direction of training “Software Engineering”, Irkutsk State Transport University

Ovsyannikov I.V. – 4th year student of the Department of Information Technologies and Information Security, direction of training “Software Engineering”, Irkutsk State Transport University

Sachkov D.I – Ph.D. in Economics, Vice-Rector for Digital Technologies, Irkutsk State Transport University, e-mail: sachkov_di@irgups.ru

Для цитирования

Коробцов В. И., Овсянников И.В., Сачков Д.И. Автоматическая генерация надежного программного кода с помощью генеративных предобученных трансформеров (GPT) // «Информационные технологии и математическое моделирование в управлении сложными системами»: электрон. науч. журн. – 2024. – №1. – С.52-59. – Режим доступа: http://ismm-irgups.ru/toma/121-2024, свободный. – Загл. с экрана. – Яз. рус., англ. (дата обращения: 17.04.2024)

For citations

Korobtsov V.I., Ovsyannikov I.V., Sachkov D.I. Automatic generation of reliable program code using generative pre-trained transformers (GPT) // Informacionnye tehnologii i matematicheskoe modelirovanie v upravlenii slozhnymi sistemami: elektronnyj nauchnyj zhurnal [Information technology and mathematical modeling in the man-agement of complex systems: electronic scientific journal], 2024. No. 1. P. 52-59. [Accessed 17/04/24]