

# Исследование применимости моделей обработки естественного языка к задаче анализа технических заданий и спецификаций на разработку программного обеспечения

Л. А. Куценко, Ю. А. Кorableв

Санкт-Петербургский государственный электротехнический университет  
«ЛЭТИ» им. В.И. Ульянова (Ленина)

lakutsenok@gmail.com, juri.korablev@gmail.com

**Аннотация.** В данной работе рассматривается применение технологий обработки естественного языка для анализа технических заданий и спецификаций в области разработки программного обеспечения. Исследование направлено на выявление потенциала NLP-моделей в автоматизации интерпретации документов, связанных с процессами разработки ПО, а также извлечения из них ключевой информации. В работе проводится обзор существующих решений, изучается теоретическая база и методология применения NLP-моделей в данной сфере. Результаты исследования демонстрируются с точки зрения их практической значимости и возможных направлений дальнейших исследований в области применения NLP для анализа технических заданий и спецификаций в разработке программного обеспечения.

**Ключевые слова:** обработка естественного языка, NLP-модель, машинное обучение, извлечение информации, планирование проектов, управление проектами, анализ технической документации, GPT

## I. ВВЕДЕНИЕ

В последнее десятилетие технологии обработки естественного языка (Natural Language Processing, сокр. NLP) претерпели значительные изменения благодаря развитию машинного обучения и искусственного интеллекта. Эти технологии нашли широкое применение во многих областях, включая разработку программного обеспечения и управления IT-проектами. Такие документы, как технические задания и спецификации, являются ключевыми источниками информации в сфере информационных технологий, определяющими требования к разрабатываемым системам, продуктам и проектам. Точный и всесторонний анализ этих документов на всех этапах ведения проектов от первичного формирования требований до разработки и внедрения конечных продуктов критически важен для успешного выполнения проектов и реализации поставленных задач.

В современных IT-компаниях, специализирующихся на разработке и поддержке программного обеспечения, функции анализа технических заданий, спецификаций и другой документации, как правило, закрепляются за системными аналитиками, руководителями проектов и другими техническими специалистами, чей ручной труд

по изучению данной документации и постановке задач требует значительных трудозатрат и не исключает возможности допущения ошибок, которые могут привести к некорректной реализации поставленных задач и увеличению сроков выполнения проектов, что, в свою очередь, приведет к финансовым потерям для компании.

Важности современных технологий NLP в сфере разработки программного обеспечения подчеркиваются в ряде исследований. Например, технологии обработки естественного языка успешно применяются для помощи при анализе и изучении программного кода [1], а также его оптимизации [2]. Эти исследования подчеркивают важность интеграции NLP в стандартные процессы разработки ПО для повышения их эффективности.

Несмотря на немалое количество исследований в данной области, остается ряд нерешенных вопросов и вызовов. Проблемы, связанные с точностью, контекстуальным пониманием и интеграцией NLP-инструментов в существующие рабочие процессы, все еще требуют дальнейшего изучения. Таким образом, данное исследование направлено на всесторонний анализ текущего состояния применения NLP в анализе спецификаций и технических заданий в сфере разработки программного обеспечения, выявление существующих проблем и определение направлений для будущих исследований.

Целью данного исследования является оценка эффективности и определение наилучших практик применения NLP в контексте анализа технических заданий и спецификаций на разработку программного обеспечения. Для реализации поставленной цели исследования требуется решить следующие задачи:

- сбор и предварительная обработка данных: необходимо провести подготовку текстовых данных из различных источников, содержащих технические задания и спецификации, а также провести предварительную обработку данных, чтобы подготовить для использования;
- выбор модели: необходимо провести анализ существующих NLP моделей для выбора наиболее подходящей к рассматриваемой задаче;
- анализ работы модели: необходимо провести оценку работу выбранной модели на

сформированном наборе данных для оценки ее применимости к рассматриваемой задаче.

## II. АВТОМАТИЗАЦИЯ АНАЛИЗА ТЕХНИЧЕСКОЙ ДОКУМЕНТАЦИИ

В настоящее время в Российской Федерации основополагающим документом, определяющим содержание технических заданий, является ГОСТ 34.602 – 2020 [3]. Согласно данному документу, техническое задание на создание информационной системы должно содержать следующие обязательные разделы:

- общие сведения – полное наименование автоматизированной системы (сокр. АС) и ее условное обозначение, шифр АС, наименование организации-заказчика разработки АС, перечень руководящих документов и другую информацию;
- цели и назначение создания автоматизированной системы;
- характеристика объектов автоматизации – сведения об условиях эксплуатации объекта автоматизации и характеристиках окружающей среды;
- требования к автоматизированной системе – требования к структуре АС в целом, требования к выполняемым функциям и задачам, общие технические требования к АС (например, стек технологий, языки программирования, требования к инфраструктуре развертывания) и другую информацию;
- состав и содержание работ по созданию автоматизированной системы – перечень этапов работ по разработке АС и их сроки;
- порядок разработки, автоматизированной систем;
- порядок контроля и приемки автоматизированной системы – виды и составы испытаний по приемке АС;
- требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие;
- требования к документированию;
- источники разработки.

Как видно из приведенного перечня, процесс анализа технической документации, который необходимо провести сотрудникам организации, ответственной за разработку программного обеспечения по поставленному техническому заданию, требует значительных временных затрат, а допущенные в нем ошибки могут привести к серьезным нарушениям в ходе выполнения работ. Кроме того, в сфере разработки ИТ-продуктов технические задания и спецификации часто представляются в произвольном виде, при этом соответствие требованиям ГОСТ могут быть не выполнены частично или полностью. Это создает дополнительные сложности при анализе технической документации и формировании задач на разработку программного обеспечения, так как формат входных требований может значительно отличаться как от заказчика к заказчику, так и от проекта к проекту.

## III. СОВРЕМЕННЫЕ ТЕХНОЛОГИИ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА

Современные средства обработки естественного языка, а в частности, крупные языковые модели (Large Language Models, сокр. LLM) благодаря обширным обучающим наборам данных и значительному увеличению вычислительной мощности, способны обрабатывать и анализировать огромные массивы текстовой информации, что делает возможным для применения в таких специфичных областях, как разработка программного обеспечения [4]. Эти модели могут выявлять сложные закономерности и зависимости в текстах, что особенно важно при работе с комплексными документами, такими как спецификациями, техническими заданиями, функциональными и нефункциональными требованиями, а также другими видами технической информации. Например, проведенное исследование применение LLM-модели в гибких методологиях разработки [5] показывают, как эффективное использование LLM может помочь в автоматизации процессов разработки программных продуктов.

Одним из ключевых преимуществ LLM является их способность к обучению на специфических наборах данных, что позволяет адаптировать их к особенностям различных доменов, включая сферу ИТ. Например, одно из исследований демонстрирует, как обучение модели на специализированных наборах данных способствует более точному проведению анализа программного кода, разработанного в ходе выполнения проектов [6].

Авторы исследований в сфере анализа естественного языка выделяют такие крупные языковые модели, как генеративный предобученный трансформер (Generative pre-trained transformer, сокр. GPT) [7], LLaMA (Large Language Model Meta AI) [8], а также основанные на них модели и расширения [9]. Эти модели демонстрируют значительный потенциал в понимании и одновременной обработке многих языков, что делает их особенно ценными для работы с документацией по программному обеспечению, так как она, во многих случаях, содержит в себе термины и определения как на русском, так и на английском языках.

Среды особенностей моделей, построенных на архитектуре трансформера (transformer), выделяют следующие факторы:

1. Архитектура трансформера позволяет обрабатывать большие объемы данных и улавливать сложные зависимости в тексте. Трансформеры используют механизмы внимания, которые позволяют модели сосредотачиваться на важных частях входных данных, что улучшает качество обучения.
2. Модели, основанные на GPT и LLaMA, являются предобученными. Это означает, что модели GPT и LLaMA уже обучены на значительных объемах данных и корпусах текста, благодаря чему они обладают широким пониманием языка и контекста, что упрощает дальнейшее обучение под конкретные задачи.

3. Данные модели поддерживают возможность дообучения (fine-tuned) для конкретных задач или доменов, используя относительно небольшие наборы данных. Это позволяет им быстро адаптироваться к новым задачам, сохраняя при этом общие обширные знания, полученные в ходе предварительного обучения.

Благодаря данным особенностям, LLM модели GPT, LLaMA и их производные были выбраны для последующего анализа технических заданий и спецификаций на разработку программного обеспечения.

#### IV. ВЫБОР МОДЕЛИ И АНАЛИЗ РЕЗУЛЬТАТОВ

Для проведения исследования и изучения возможности автоматизации анализа технических заданий и спецификаций была выбрана модель OpenAI GPT-4 [10], показывающая наилучшие результаты в задачах из различных сфер [11]. В будущих исследованиях по данной тематике предполагается применение других LLM моделей, не имеющих ограничений на коммерческое использование, а также свободное встраиваемых в целевое ПО (проекты с открытым исходным кодом). Однако, в контексте исследования возможности применения данной технологии к рассматриваемой задаче анализа технической документации, модель GPT-4 является допустимым выбором.

На момент проведения исследования авторами не было обнаружено русскоязычного набора данных технических заданий и спецификаций на разработку программного обеспечения в открытом доступе. В связи с этим, для формирования набора тестовых данных авторами был подготовлен соответствующий набор данных, основанный на открытой информации, доступной в Единой информационной системе в сфере закупок (ЕИС) Российской Федерации [12]. Для формирования набора данных были выбраны 100 случайных конкурсных процедур, найденных в ЕИС при помощи механизма фильтрации со следующими входными данными: закон – 44-ФЗ, этап закупки – закупка завершена, код ОКПД2 [13] – 62.01.1: Услуги по проектированию, разработке информационных технологий для прикладных задач и тестированию программного обеспечения

В качестве документа, являющимся техническим заданием конкурсной процедуры, было использовано описание объекта закупки, в котором, в соответствии с частью 2 статьи 33 44-ФЗ [14], «указываются функциональные, технические и качественные характеристики, эксплуатационные характеристики объекта закупки». Загруженные документы были размечены вручную, в результате чего по каждому из них были выделены некоторые характеристики информационных систем, сформированные на основе ГОСТ 34.602 – 2020, которые будут использованы для извлечения средствами рассматриваемой модели, а именно:

- наименование;
- цели и назначения разработки;
- описание структуры;

- стек технологий;
- требования к инфраструктуре;
- требования к документированию.

В ходе анализа, полученного набор данных из 100 документов, были обнаружены следующие расширения документов с описанием объекта закупки: doc/docx (Microsoft Word), odt (Open Document), rtf (Rich Text Format), pdf, а также zip-архивы, содержащие один или несколько документов указанных форматов. Для стандартизации данных документов и приведения их в простой текстовый вид, используемый для формирования запросов к GPT-моделям, был реализован скрипт-обработчик на языке программирования Python.

1. Для извлечения информации из документов с расширениями doc, docx и odt использовалась библиотека ruandoc [15], позволяющие извлечь текстовые данные с сохранением порядка и форматирования исходный блоков документов.
2. Для обработки pdf скан-образов закупочной документации использовалась библиотека tesseract [16], выполняющая алгоритм оптического распознавания символов (optical character recognition, сокр. OCR) страниц исходного документа, после из разных страниц текст объединялся с сохранением исходного порядка.
3. Для обработки zip-архивов была использована библиотека zipfile [17], с помощью которой производилась разархивация исходных файлов и последующая обработка каждого документа в зависимости от его расширения по алгоритмам, описанным в пунктах 1 и 2. Полученные текстовые блоки были объединены в единые текстовые документы, так как во всех обнаруженных документах необходимость использования архивов была обусловлено выделением приложений к основному документу в отдельные файлы.

Работа с моделями, такими как GPT-4, основана на формировании запросов (промтов, от англ. – prompt) на естественном языке. Правильное формирование промптов к моделям LLM является ключевым аспектом для корректного извлечения информации и задания команд. Авторы других исследований в сфере отмечают важность указания целевой задачи, контекста (например, сферы деятельности), роли (актера), а также формата ответа для получения наиболее точных ответов на запросы [18]. Для проведения исследования модели GPT-4 был использован режим чата, когда модель запоминает контекст предыдущих запросов, что использовалось для первичной загрузки текста технического задания в модель и последующего формирования промтов. Например, для загрузки технического задания использовался промт:

Человек: *“Ты являешься системным аналитиком в сфере разработке программного обеспечения. Твоя задача – запомнить текст технического задания, описанного ниже, для последующего анализа. Текст задания: ... “*

Для последующего анализа рассмотрим конкурсную процедуру № 0848300051723000350 на «оказание услуг по созданию информационного онлайн-ресурса (веб-сайта) инвестиционного портала Сергиево-Посадского городского округа Московской области» [19], включенную в исходный набор данных. После загрузки текста технического задания модели был задан запросы по каждому из извлекаемых параметров, например:

1. Человек: *«На основе ранее загруженного текста технического задания сформируй список требуемых для реализации технологий. Выведи результат в виде кратких названий технологий, разделенных запятой.»*

Модель GPT-4: *«IC-Битрикс, HTML, PHP, HTML5, CSS3.»*

2. Человек: *«На основе ранее загруженного текста технического задания укажи только полное наименование разрабатываемой системы.»*

Модель GPT-4: *«Информационный онлайн-ресурс (веб-сайт) инвестиционного портала Сергиево-Посадского городского округа Московской области.»*

3. Человек: *«На основе ранее загруженного текста технического задания напиши список этапов работ с указанием названия этапа, планов даты начала и окончания, а также сроков выполнения.»*

Модель GPT-4:

- Покупка лицензии программы «IC-Битрикс: Управление сайтом»: 01 марта 2024 года – 30 марта 2024 года (30 календарных дней).
- Разработка и согласование структуры сайта: 31 марта 2024 года – 29 апреля 2024 года (30 календарных дней).
- Разработка «Главной страницы»: 30 марта 2024 года – 14 мая 2024 года (45 календарных дней).
- Разработка страницы «Инвестору»: 15 мая 2024 года – 29 июня 2024 года (45 календарных дней)

Последний пример особенно интересен тем, что в исходном документе указаны только сроки выполнения этапов в календарных днях, при этом плановый срок начала работ указан в другом разделе. Следовательно, модель выполнила верный расчет дат и сроков реализации этапов.

Результаты, полученные в ходе анализа работы модели GPT-4 на созданном наборе данных по представленным параметрам, представлены в табл. 1. Корректность ответов по каждому параметру оценивалась как среднее значение корректности по каждому экземпляру набора данных. Корректность параметра экземпляра набора данных оценивалась по общему набору ожидаемых характеристик ответа с помощью стандартных метрик, таких как точность, точность (precision), полнота (recall) и F1-мера.

ТАБЛИЦА I. РЕЗУЛЬТАТЫ АНАЛИЗА РАБОТЫ МОДЕЛИ GPT-4

	Характеристика	F1
1	Наименование	0.89
2	Цели и назначения разработки	0.86
3	Описание структуры	0.69
4	Стек технологий	0.75
5	Требования к инфраструктуре	0.79
6	Требования к документированию	0.77

Как видно по представленной таблице, модель показывает точность работы 80.3 % как среднее значение по рассматриваемым характеристикам. Однако, важно отметить, что несмотря на высокую точность работы модели и возможность ее применения к задаче автоматизированного анализа технических заданий, для дальнейшего повышения точности следует использовать модели, позволяющие проводить дообучение (fine-tuning), для повышения точности понимания контекст конкретных сфер и областей работы, таких как разработка ПО.

## V. ЗАКЛЮЧЕНИЕ

В данной статье было представлено исследование применимости моделей GPT-4 к задаче анализа технических заданий и спецификаций на разработку программного обеспечения, результаты которого показали, что исходная модель может обрабатывать тексты предоставленных документов с точностью 80.3 %, что делает возможным применение модели GPT-4 к рассматриваемой задаче, что в будущем позволит автоматизировать процесс анализа технических заданий в сфере разработки программного обеспечения.

Дальнейшее развитие и оптимизация исходной, а также использование моделей с открытым исходным кодом, позволяющие проанализировать дообучение для работы в специфических сферах, таких как разработка программных продуктов, могут открыть новые возможности и применения в будущих исследованиях.

## СПИСОК ЛИТЕРАТУРЫ

- [1] Guo Q., Cao J., Xie X., Liu S., Li X., Chen B., & Peng X. Exploring the potential of chatgpt in automated code refinement: An empirical study // Proceedings of the 46th IEEE/ACM International Conference on Software Engineering. 2024. С. 1-13.
- [2] AlOmar E.A., Venkatakrisnan A., Mkaouer M.W., Newman C.D., & Ouni A. How to Refactor this Code? An Exploratory Study on Developer-ChatGPT Refactoring Conversations //arXiv preprint arXiv:2402.06013. 2024.
- [3] ГОСТ 34.602-2020. Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. М.: Российский институт стандартизации, 2021.
- [4] Liu Y., Han T., Ma S., Zhang J., Yang Y., Tian J. Summary of chatgpt-related research and perspective towards the future of large language models // Meta-Radiology. 2023. С. 100017.
- [5] Bera P., Wautelet Y., Poels G. On the Use of ChatGPT to Support Agile Software Development //Short Paper Proceedings of the Second International Workshop on Agile Methods for Information Systems Engineering (Agil-ISE 2023) co-located with the 35th International Conference on Advanced Information Systems Engineering (CAiSE 2023), Zaragoza, Spain, June 13, 2023. CEUR-WS. org, 2023. Т. 3414. С. 1-9.
- [6] Pornprasit C., Tantithamthavorn C. GPT-3.5 for Code Review Automation: How Do Few-Shot Learning, Prompt Design, and Model Fine-Tuning Impact Their Performance? //arXiv preprint arXiv:2402.00905. 2024.

- [7] Radford A., Narasimhan K., Salimans, T., & Sutskever I. (2018). Improving language understanding by generative pre-training.
- [8] Touvron H., Lavril T., Izacard G., Martinet X., Lachaux M.A., Lacroix T. Llama: Open and efficient foundation language models //arXiv preprint arXiv:2302.13971. 2023.
- [9] Hou X., Zhao Y., Liu Y., Yang Z., Wang K., Li L., Luo X., Lo D., Grundy J., Wang H. Large language models for software engineering: A systematic literature review //arXiv preprint arXiv:2308.10620. 2023.
- [10] OpenAI, Gpt-4 technical report //arXiv preprint arXiv:2303.08774. 2023.
- [11] Hadi M.U., Qureshi R., Shah A., Irfan M., Zafar A., Shaikh M. A survey on large language models: Applications, challenges, limitations, and practical usage // Authorea Preprints. 2023.
- [12] Официальный сайт Единой информационной системы в сфере закупок [Электронный ресурс]. URL: <https://zakupki.gov.ru/epz/main/public/home.html> (дата обращения 01.03.2024).
- [13] ОКПД 2 – Общероссийский классификатор продукции по видам экономической деятельности. Классификатор ОКПД 2 ОК 034-2014 (КПЕС 2008) с изменением №99 от 1 марта 2024 г. [Электронный ресурс]. URL: <https://classifikators.ru/okpd> (дата обращения 01.03.2024).
- [14] Федеральный закон от 05.04.2013 № 44-ФЗ «О контрактной системе в сфере закупок товаров, работ, услуг для обеспечения государственных и муниципальных нужд» [Электронный ресурс]. URL: <http://publication.pravo.gov.ru/Document/View/0001201304080023> (дата обращения 01.03.2024).
- [15] ruandoc 1.13 URL: <https://pypi.org/project/ruandoc/> (дата обращения 01.03.2024).
- [16] Patel C., Patel A., Patel D. Optical character recognition by open-source OCR tool tesseract: A case study // International Journal of Computer Applications. 2012. Т. 55. №. 10. С. 50-56.
- [17] zipfile – Work with ZIP archives¶ URL: <https://docs.python.org/3/library/zipfile.html> (дата обращения 01.03.2024).
- [18] Liu X., Zheng Y., Du Z., Ding M., Qian Y., Yang Z., & Tang J. GPT understands, too //AI Open. 2023.
- [19] Оказание услуг по созданию информационного онлайн-ресурса (веб-сайта) инвестиционного портала Сергиево-Посадского городского округа Московской области. Единый портал государственных закупок. URL: <https://zakupki.gov.ru/epz/order/notice/ea20/view/common-info.html?regNumber=0848300051723000350> (дата обращения 01.03.2024).