

Research of Applicability of Natural Language Processing Models to the Task of Analyzing Technical Tasks and Specifications for Software Development

Leonid A. Kutsenok

*Faculty of Computer Science and Technologies
Saint Petersburg Electrotechnical University "LETI"
Saint Petersburg, Russia
lakutsenok@gmail.com*

Yuri A. Korablev

*Faculty of Computer Science and Technologies
Saint Petersburg Electrotechnical University "LETI"
Saint Petersburg, Russia
juri.korablev@gmail.com*

Abstract— This paper examines the application of natural language processing technologies for analyzing technical tasks and specifications in the field of software development. The research is aimed at identifying the potential of NLP models in automating the interpretation of documents related to software development processes, as well as extracting key information from them. The paper reviews existing solutions, explores the theoretical basis and methodology for applying NLP models in this area. The results of the study are demonstrated in terms of their practical significance and possible directions for further research in the field of NLP application for analyzing technical tasks and specifications in software development.

Keywords—*Natural Language Processing, NLP model, Machine Learning, Information Extraction, Project Planning Technical Documentation Analysis, GPT*

I. INTRODUCTION

In the last decade, Natural Language Processing (NLP) technologies have undergone significant changes due to the development of machine learning and artificial intelligence. These technologies have found wide application in many fields, including software development and IT projects. Documents such as terms of reference and specifications are key sources of information in the IT industry, defining the requirements for the systems, products and projects being developed. Accurate and comprehensive analysis of these documents at all stages of IT-projects, from initial requirements generation to the development and implementation of final products, is critical to successful project execution and delivery.

In modern IT-companies specializing in software development and support, the functions of analyzing technical tasks, specifications and other documentation are usually assigned to system analysts, project managers and other technical specialists, whose manual work on studying this documentation and setting tasks requires considerable labor input and does not exclude the possibility of making mistakes that can lead to incorrect implementation of the tasks and increase the project implementation time, which, in turn, will lead to financial losses for the company.

The importance of modern NLP technologies in the field of software development has been emphasized in a number of studies. For example, natural language processing techniques have been successfully applied to help analyze and learn software code [1], as well as optimize it [2]. These studies emphasize the importance of integrating NLP into standard software development processes to improve their efficiency.

Despite the considerable amount of research in this area, there are still a number of unresolved issues and challenges. Issues related to accuracy, contextual understanding and integration of NLP tools into existing workflows still require further research. Thus, this study aims to comprehensively analyze the current state of application of NLP in the analysis of specifications and statements of work in software engineering, identify existing problems and identify directions for future research.

The purpose of this study is to evaluate the effectiveness and identify best practices for applying NLP in the context of analyzing software development terms of reference and specifications. In order to realize the research objective, the following tasks are required:

- Data collection and pre-processing: text data from various sources containing terms of reference and specifications should be prepared, and data should be pre-processed to prepare for use;
- Model selection: it is necessary to analyze existing NLP models to select the most appropriate one for the task at hand;
- Model performance analysis: it is necessary to evaluate the performance of the selected model on the generated dataset to assess its applicability to the problem at hand.

II. AUTOMATION OF TECHNICAL DOCUMENTATION ANALYSIS

Currently, in the Russian Federation, the fundamental document defining the content of technical specifications is GOST 34.602 – 2020 [3]. According to this document, the terms of reference for the creation of an information system should contain the following mandatory sections:

- General information – the full name of the automated system and its conventional designation, the code of the system, the name of the organization-customer of the system development, the list of guiding documents and other information;
- The goals and purpose of creating an automated system;
- Characterization of automation objects - information about the operating conditions of the automation object and environmental characteristics;
- Requirements to the automated system - requirements to the structure of the system as a whole, requirements

to the functions and tasks to be performed, general technical requirements to the automated system (e.g., technology stack, programming languages, deployment infrastructure requirements) and other information;

- Composition and content of works on creation of the automated system - list of stages of works on AS development and their terms;
- Order of development, automated systems;
- Order of control and acceptance of the automated system types and composition of tests for the acceptance of the AC;
- Requirements for the composition and content of work to prepare the automation object for putting the automated system into operation;
- Documentation requirements;
- Sources of development.

As can be seen from the above list, the process of analysis of technical documentation, which should be carried out by the employees of the organization responsible for software development according to the delivered technical task, requires considerable time expenditures, and mistakes made in it can lead to serious violations in the course of work performance. Besides, in the sphere of IT-products development technical tasks and specifications are often provided in an arbitrary form, and compliance with GOST requirements may not be fulfilled partially or completely. This creates additional difficulties when analyzing technical documentation and forming tasks for software development, as the format of input requirements may differ significantly both from customer to customer and from project to project.

III. MODERN APPROACHES TO NATURAL LANGUAGE PROCESSING

Modern natural language processing tools and, in particular, Large Language Models (LLM), due to extensive training datasets and significantly increased computational power, are capable of processing and analyzing huge amounts of textual information, which makes it possible to apply in such specific areas as software development [4]. These models can identify complex patterns and dependencies in texts, which is especially important when dealing with complex documents such as specifications, statements of work, functional and non-functional requirements, and other types of technical information. For example, a study conducted on the application of LLM model in agile development methodologies [5] shows how the effective use of LLM can help in automating software product development processes.

One of the key advantages of LLMs is their ability to be trained on specific datasets, which allows them to be adapted to the peculiarities of different domains, including the IT domain. For example, one study demonstrates how training a model on specialized datasets facilitates more accurate analysis of software code developed during project execution [6].

Authors of research in the field of natural language analysis emphasize such large language models as Generative pre-trained transformer (GPT) [7], LLaMA (Large Language Model Meta AI) [8], and models and extensions based on them [9]. These models show significant potential in understanding and simultaneous processing of

many languages, which makes them especially valuable for working with software documentation, since it, in many cases, contains terms and definitions in both Russian and English.

The following factors stand out among the features of models based on the transformer architecture:

1. Transformer architecture can handle large amounts of data and capture complex dependencies in text. Transformers utilize attention mechanisms that allow the model to focus on important parts of the input data, which improves learning.
2. GPT and LLaMA based models are pre-trained. This means that GPT and LLaMA models have already been trained on significant amounts of data and text corpora, so they have a broad understanding of language and context, which makes it easier to further train them for specific tasks.
3. These models support the ability to fine-tuned for specific tasks or domains using relatively small datasets. This allows them to quickly adapt to new tasks while maintaining the overall extensive knowledge gained during pre-training.

Because to these features, the LLM models GPT, LLaMA and their derivatives were selected for subsequent analysis of software development terms of reference and specifications.

IV. MODEL SELECTION AND RESULT ANALYSIS

OpenAI GPT-4 model [10], which shows the best results in tasks from different spheres [11], was chosen for the research and studying the possibility of automating the analysis of technical tasks and specifications. In future studies on this topic it is supposed to use other LLM models that have no restrictions on commercial use, as well as freely embedded in the target software (open source projects). However, in the context of studying the possibility of applying this technology to the considered task of analyzing technical documentation, the GPT-4 model is an acceptable choice.

At the time of the study, the authors did not find a Russian-language dataset of technical tasks and specifications for software development in the public domain. In this regard, to form the test data set, the authors prepared an appropriate data set based on open information available in the Unified Information System in the field of procurement (UIS, Russian – ЕИЦ) of the Russian Federation [12]. For the formation of the data set were selected 100 random competitive procedures found in the UIS using the filtering mechanism with the following input data: law – 44-FZ (Russian – Федеральный закон, ФЗ), procurement stage – procurement completed, OKPD2 code [13] – 62.01.1: Services on design, development of information technologies for applied tasks and software testing.

As a document, which is the terms of reference of the tender procedure, was used the description of the procurement object, which, in accordance with Part 2 of Article 33 of 44-FZ [14], "indicates the functional, technical and quality characteristics, operational characteristics of the procurement object". The uploaded documents were marked manually, as a result of which for each of them were allocated some characteristics of information systems, formed on the basis of GOST 34.602 – 2020, which will be used for extraction by means of the model under consideration, namely:

- Naming;
- Goals and purposes of the development;
- Structure Description;
- Technology Stack;
- Infrastructure requirements;
- Documentation requirements.

In the course of the analysis, obtained dataset of 100 documents, the following extensions of documents describing the subject of the procurement were found: doc/docx (Microsoft Word), odt (Open Document), rtf (Rich Text Format), pdf, as well as zip-archives containing one or more documents of the specified formats. In order to standardize these documents and bring them into a simple text form used for forming queries to GPT models, a script handler in the Python programming language was implemented.

1. The py pandoc library [15] was used to extract information from documents with doc, docx and odt extensions, which allow to extract text data while preserving the order and formatting of the original document blocks.
2. To process pdf scans of procurement documents we used the library tesseract [16], which performs the algorithm of optical character recognition (OCR) of the pages of the source document, after which the text from different pages was combined with preserving the original order.
3. To process zip-archives, the zipfile library [17] was used to unzip the source files and then process each document depending on its extension according to the algorithms described in items 1 and 2. The obtained text blocks were merged into single text documents, since in all the detected documents the necessity of using archives was conditioned by the separation of appendices to the main document into separate files.

Working with models such as GPT-4 is based on the generation of natural language prompts. The correct formation of Prompts to LLM models is a key aspect for correct information retrieval and command assignment. Authors of other studies in the field note the importance of specifying the target task, context (e.g., domain), role (actor), and response format to obtain the most accurate responses to queries [18]. For the GPT-4 model study, a chat mode was used where the model remembers the context of previous queries, which was used to initially load the text of the terms of reference into the model and subsequently generate prompts. For example, the prompt was used to load the terms of reference:

Human: *"You are a system analyst in software development. Your task is to memorize the text of the statement of work described below for later analysis. The text of the document: ..."*

For further analysis let us consider the competitive procedure № 0848300051723000350 for "provision of services for the creation of an online information resource (website) of the investment portal of Sergiev Posadsky urban district of Moscow region" [19], included in the initial dataset. After loading the text of the model's terms of reference, queries were set for each extracted parameters, e.g (prompts and answers are translated from Russian into English):

1. Human: "Based on the previously loaded text of the terms of reference, generate a list of technologies required for implementation. Output the result in the form of short names of technologies separated by a comma."

GPT-4 model: *"1C-Bitrix, HTML, PHP, HTML5, CSS3."*

2. Human: "Based on the previously uploaded text of the terms of reference, specify only the full name of the system to be developed".

GPT-4 Model: *"Online information resource (website) of the investment portal of Sergiev Posadsky Urban District of the Moscow Region".*

3. Human: "Based on the previously uploaded text of the terms of reference, write a list of work phases with the name of the phase, planned start and end dates, and due dates."

GPT-4 Model:

- *Purchase of 1C-Bitrix: Site Management" software license: March 01, 2024 - March 30, 2024 (30 calendar days).*
- *Development and coordination of the site structure: March 31, 2024 - April 29, 2024 (30 calendar days).*
- *Development of the "Home Page": March 30, 2024 - May 14, 2024 (45 calendar days).*
- *Development of the "Investor" page: May 15, 2024 - June 29, 2024 (45 calendar days)*

The last example is particularly interesting because the source document only specifies milestone dates in calendar days, while the planned start date is specified in another section. Therefore, the model has correctly calculated the dates and deadlines of the milestones.

The results obtained by analyzing the performance of the GPT-4 model on the created dataset for the presented parameters are presented in Table 1. The correctness of the responses for each parameter was evaluated as the average correctness value for each dataset instance. The parameter correctness of a dataset instance was evaluated against a common set of expected response characteristics using standard metrics such as precision, accuracy (precision), completeness (recall), and F1-measure.

TABLE I. RESULTS OF GPT-4 MODEL

	Characteristic	F1
1	Naming	0.89
2	Goals and purposes of the development	0.86
3	Structure Description	0.69
4	Technology Stack	0.75
5	Infrastructure requirements	0.79
6	Documentation requirements	0.77

As can be seen from the table below, the model shows an accuracy of 80.3% as an average value for the considered characteristics. However, it is important to note that despite the high accuracy of the model and the possibility of applying it to the task of automated analysis of technical specifications, to further improve the accuracy, models that allow fine-tuning should be used to improve the accuracy of understanding the context of specific domains and areas of work, such as software development.

V. CONCLUSION

This paper presented study of applicability of GPT-4 models to the task of analyzing technical tasks and specifications for software development, the results of which showed that the original model processes the texts of the provided documents with an accuracy of 80.3%, which makes it possible to apply the GPT-4 model to the considered task, which in the future will allow to automate the process of analyzing technical tasks in the field of software development.

Further development and optimization of the original, as well as the use of open-source models that allow fine-tuning for specific areas, such as software product development, may open new opportunities and applications in future studies

REFERENCES

- [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. P. 1-13.
- [2] AlOmar, E.A., Venkatakrishnan, 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] State Standart 34.602-2020. Information technologies. Complex of standards for automated systems. Terms of reference for the creation of an automated system. Moscow: Russian Institute of Standardization, 2021. (In Russian)
- [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. P. 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. V. 3414. P. 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] Official website of the Unified Information System in the field of procurement. URL: <https://zakupki.gov.ru/epz/main/public/home.html> (referenced on 01.03.2024).
- [13] OKPD2 – All-Russian Classifier of Products by Economic Activities. Classifier OKPD 2 OK 034-2014 (KPES 2008) with change No. 99 of March 1, 2024. URL: <https://classifikators.ru/okpd> (referenced on 01.03.2024).
- [14] Federal Law of 05.04.2013 No. 44-FZ "On the contractual system in the sphere of procurement of goods, works, services for state and municipal needs". URL: <http://publication.pravo.gov.ru/Document/View/0001201304080023> (referenced on 01.03.2024).
- [15] pypandoc 1.13 URL: <https://pypi.org/project/pypandoc/> (referenced on 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. V. 55. №. 10. P. 50-56.
- [17] zipfile — Work with ZIP archives URL: <https://docs.python.org/3/library/zipfile.html> (referenced on 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] Provision of services for the creation of an online information resource (website) of the investment portal of Sergiev Posadsky urban district of Moscow region. URL: <https://zakupki.gov.ru/epz/order/notice/ea20/view/common-info.html?regNumber=0848300051723000350> (referenced on 01.03.2024).