NTCIR 17 Conference: Proceedings of the 17th NTCIR Conference on Evaluation of Information Access Technologies, December 12-15, 2023, Tokyo, Japan
DOI: https://doi.org/10.20736/0002001283

Zero-shot classification of TNM staging for Japanese radiology report using ChatGPT at RR-TNM subtask of NTCIR-17 MedNLP-SC

Mizuho Nishio †
Center for Advanced Medical Engineering Research & Development
Kobe University
Kobe, Japan
nishiomizuho@gmail.com

Hidetoshi Matsuo
Department of Radiology
Kobe University
Kobe, Japan
yukikaze.jp@gmail.com

Takaaki Matsunaga
Department of Radiology
Kobe University
Kobe, Japan
takaaki.matsunaga@live.jp

Koji Fujimoto
Advanced Imaging in Medical Magnetic Resonance
Kyoto University
Kyoto, Japan
kfb@kuhp.kyoto-u.ac.jp

Morteza Rohanian
Department of quantitative Biomedicine
University of Zurich
Zurich, Switzerland
m.rohanian@qmul.ac.uk

Farhad Nooralahzadeh
Department of quantitative Biomedicine
University of Zurich
Zurich, Switzerland
Farhad.Nooralahzadeh@uzh.ch

Fabio Rinaldi
Dalle Molle Institute for Artificial Intelligence
Lugano, Switzerland
fabio.rinaldi@idsia.ch

Michael Krauthammer
Department of quantitative Biomedicine
University of Zurich
Zurich, Switzerland
michael.krauthammer@uzh.ch

ABSTRACT

We describe our submission to the RR-TNM subtask of the NTCIR-17 MedNLP-SC shared task. In the RR-TNM subtask, we developed our system for automatic extraction and classification of the TNM staging from Japanese radiology reports of lung cancers. In our system, zero-shot classification and prompt engineering were performed using ChatGPT and LangChain, respectively. According to the accuracies calculated by the organizers of the RR-TNM subtask, the accuracies of N and M factors in the TNM staging were higher in our submission than in the other submissions. These results indicate that our system with ChatGPT and LangChain may be promising.

KEYWORDS

Medical Natural Language Processing, Radiology Reports, Lung Cancer, Cancer Staging, TNM Staging, ChatGPT, Zero-shot Classification, LangChain, Prompt Engineering.

TEAM NAME

KRad

SUBTASKS

RR-TNM

1 INTRODUCTION

Radiologists read a large number of images in their daily clinical practice. The results of radiologists' evaluations are recorded as radiology reports. From radiology examinations, a large number of images are generated, leading to a significant number of reports as well. In Japanese hospitals, most radiology reports are written for CT and MRI examinations. The radiology reports can be a valuable source of information to improve clinical practice and support clinical research. However, radiology reports are recorded in natural language as unstructured data, making it difficult to efficiently process such a large volume of reports manually. Radiology Report TNM Staging (RR-TNM) subtask was performed as a part of the NTCIR-17 Medical Natural Language Processing for Social Media and Clinical Texts (MedNLP-SC) shared task for overcoming this problem [1].

In the RR-TNM subtask, radiology reports for CT examination of lung cancer are the targets to be analyzed. The RR-TNM subtask provides an open benchmark of radiology reports. Researchers of radiology mainly focus on imaging, and

radiology reports are not common as the target of radiology research. As a result, it is challenging to use radiology reports for research in the field of radiology. Therefore, the RR-TNM subtask is a unique opportunity for researchers of natural language processing (NLP) using radiology reports.

1.1 Lung cancer

Lung cancer is one of the most common cancers worldwide and is a major cause of cancer-related deaths [2]. There are two major types of lung cancers. (i) Non-small cell lung cancer is the most common type, accounting for about 85% of all lung cancers. Non-small cell lung cancer includes several subtypes, such as adenocarcinoma, squamous cell carcinoma, and large cell carcinoma. (ii) Small cell lung cancer is less common but tends to grow and spread more aggressively than non-small cell lung cancer. Diagnosis of lung cancer typically involves imaging examinations such as chest X-ray, CT, and 18F-fluoro-2-deoxy-D-glucose PET, along with biopsy samples taken from the lung tissue to confirm cancerous cells.

1.2 TNM staging of lung cancer

Cancer stage is one of the most important prognostic parameters in most types of cancers. The Union for International Cancer Control (UICC) specifies criteria for staging each type of cancers including lung cancer [3]. TNM staging of lung cancer is a system used to describe the extent of cancer growth and spread. The same TNM staging system is used for both small cell lung cancer and non-small cell lung cancer. The TNM staging system involves three key factors: T (tumor size and invasiveness), N (lymph node involvement), and M (metastasis to other parts of the body). The TNM staging is determined by combining these factors to provide a comprehensive picture of the cancer progression.

For readers not familiar with the TNM staging of lung cancer, a simplified version of T, N, and M factors of UICC version 8 are described below. The details of the TNM staging of UICC version 8 are available elsewhere [3,4].

T factor (simplified version)

- T1: Size of lung cancer, <3 cm
- T2: Size of lung cancer, 3-5 cm
- T3: (Size of lung cancer, 5-7 cm) or (Local invasion of chest wall, parietal pericardium, phrenic nerve)
- T4: (Size of lung cancer, >7 cm) or (Invasion to mediastinum, trachea, heart/great vessels, esophagus, vertebra, carina, recurrent laryngeal nerve)

N factor (simplified version)

- N0: No regional lymph node metastasis
- N1: Metastasis in ipsilateral peribronchial and/or hilar lymph node and intrapulmonary node
- N2: Metastasis in ipsilateral mediastinal and/or subcarinal lymph nodes
- N3: Metastasis in contralateral mediastinal, contralateral hilar, ipsilateral, or contralateral scalene, or supraclavicular lymph node(s)

M factor (simplified version)

- M0: No distant metastasis
- M1: Distant metastasis

Information related to the TNM staging is typically recorded in clinical narrative text. For example, for images of CT and PET, radiologists frequently record results of the TNM staging of lung cancer using natural language as radiology reports. As a result, for accurately extracting the staging information, it is necessary to search a large amount of text and resolve discordance between different data sources.

1.3 Automatic extraction and classification of TNM staging

Because information related to the TNM staging is typically recorded as unstructured text, human chart-abstractors must check narrative text to accurately extract the staging information. However, extracting and classifying the TNM staging from unstructured text manually can be time-consuming and error-prone. On the other hand, automatic extraction and classification of the TNM staging from unstructured text can provide several significant benefits in the medical and healthcare domains. For example, (i) the system with automatic extraction and classification of the TNM staging can save valuable time for healthcare professionals and researchers, allowing them to focus on more critical tasks. (ii) The automated systems can significantly enhance the accuracy of extracting and classifying information of the TNM staging information from vast amounts of unstructured text data. (iii) Large-scale text mining and analysis of the TNM staging can facilitate comprehensive research studies. Researchers can identify trends, patterns, and correlations within the data, leading to new insights and advancements in cancer treatment and care.

1.4 Purpose of the present study

The purpose of the present study is to automatically extract and classify the TNM staging from Japanese radiology reports of lung cancers. Our previous study [5] of Subtask3-RR-JA (CI-RR-JA) in NTCIR-16 RealMed-NLP [6] extracted and used information related to the TNM staging of lung cancer. While in our previous studies of NTCIR-16 [5,7] we fine-tuned Japanese pre-trained masked-language model for the extraction, in the present study we used a large language model (LLM), ChatGPT, for zero-shot classification (without fine-tuning) of the TNM staging.

2 RELATED WORKS

2.1 NLP for radiology report

NLP is a set of techniques that involve the processing of human natural language by computers. NLP plays a significant role in the advancement of medicine and has been shown to be important in extracting structured information from radiology reports. Traditionally, rule-based methods of NLP have been employed. Recent advances in NLP have been accompanied by applications of machine learning and deep learning.

In comparison to computer vision in radiology [8,9], the artificial intelligence (AI) community in radiology has not extensively explored NLP. For instance, in one review paper, it is stated that recent reviews on NLP application to radiology are limited [10]. However, NLP is expected to be useful for reducing the workload of radiologists by extracting structured information from radiology reports.

2.2 NLP for automatic extraction and classification of TNM staging of lung cancer

As applications of NLP for automatic extraction and classification of the TNM staging, there are several previous studies. Hu et al. aimed to automatically extract the staging-related information from CT reports of lung cancer [11]. Their

deep learning model achieved the best performance with a macro-F1 score of 97.13% and a micro-F1 score of 98.37%. Gupta et al. developed the system for automatic classification of the TNM staging using the rule-based method along with deep learning technology called long short-term memory [12]. Their empirical experiment shows the accuracy of up to 85%. AAlAbdulsalam et al. proposed novel applications of NLP and machine learning to automatically extract and classify TNM stage mentions from records at the Utah Cancer Registry [13]. Their results indicate that the TNM stages can be automatically extracted and classified with high accuracy (extraction sensitivity: 95.5%‑98.4% and classification sensitivity: 83.5%‑87%).

2.3 Large language models

LLM has a relatively short but significant history in NLP. A major breakthrough occurred with the advent of deep learning techniques and the development of neural network-based models, such as recurrent neural networks and long short-term memory networks. The real revolution came with the introduction of transformer-based architectures, such as Bidirectional Encoder Representations from Transformers (BERT) [7] and Generative Pre-trained Transformer (GPT) [14‑16]. These LLMs enabled impressive achievements in NLP tasks, including language translation, sentiment analysis, question answering, and text generation. They have also been adapted for a wide range of applications, from chatbots to content creation, and have become a crucial part of AI research and development.

2.4 ChatGPT

ChatGPT is based on the GPT architecture developed by OpenAI. The development of ChatGPT began with the original GPT-1 model, released in 2018, which demonstrated the power of transformer-based language models for various language tasks [14]. Subsequent iterations brought significant advances in terms of model size, performance, and capabilities. GPT-2, introduced in 2019, was a notable breakthrough due to its impressive ability to generate coherent and contextually relevant text [15]. In June 2020, OpenAI released GPT-3, a giant model with 175 billion parameters, marking a new milestone in the field of NLP [16]. GPT-3's capabilities went beyond previous iterations, demonstrating remarkable capabilities in language understanding, question answering, and even creative text generation. Following the success of GPT-3, OpenAI introduced ChatGPT, a user-friendly version designed for interactive conversations with users [17]. It was made available to the public through the ChatGPT website and its application programming interface (API), making it accessible to developers and businesses for various applications such as chatbots, customer support, and content generation.

2.5 Zero-shot classification

Zero-shot classification is a machine learning paradigm in which a model is capable of recognizing and classifying objects or concepts that it has never seen during training [16]. Instead of relying on direct examples, it uses auxiliary information such as attributes or semantic embeddings to generalize to unseen classes. This approach enables the model to classify new categories without explicit data for each category making it more flexible and adaptable in real-world scenarios. In LLMs such as GPT3 and ChatGPT, the capability of zero-shot classification is high compared to other conventional NLP models.

2.6 Prompt engineering

Prompt engineering is a technique in NLP that involves using example prompts to instruct LLMs, such as ChatGPT, to perform specific tasks or generate desired outputs [18]. The prompt engineering does not always require fine-tuning of the LLM. Through the prompt engineering and/or custom prompts, the LLM can be adapted to different domains or applications, making it more versatile and effective for various tasks. This approach allows the model to be more controlled in its responses, improving its ability to provide accurate and contextually relevant answers. For the prompt engineering and/or the custom prompts, the dedicated library (LangChain [19]) is useful.

3 METHODS

In this section, we describe our system to utilize ChatGPT (GPT version 3.5 (gpt3.5-turbo)) for extracting and classifying the TNM staging from Japanese radiology reports. For the development of our system, python (version 3.10) and LangChain (version 0.0.180) were used. To utilize ChatGPT in our system, the OpenAI's API was used through our python script and openai python package (version 0.27.7). In our system, the fine-tuning was not used. Therefore, zero-shot classification of the TNM staging was performed in our system.

3.1 Dataset of Japanese radiology reports

The details and the method to build the RR-TNM dataset are described in the task overview paper by the organizers of the RR-TNM subtask [1]. In this subsection, the RR-TNM dataset is briefly described. The RR-TNM dataset was labeled under TNM criteria of the 8th edition by the Japan Lung Cancer Society, which is compatible with the UICC criteria. Radiopaedia (https://radiopaedia.org/), an open-access radiology reference, was used as the source of lung cancer cases, and 27 cases from Radiopaedia were included as lung cancer cases in the RR-TNM dataset. Then, nine Japanese board-certified radiologists generated free-text radiology reports for each case. Finally, all stage labels were assigned to the radiology reports based on those of the nine radiologists' reports. In some cases, manual correction was performed during the labeling process. For the RR-TNM subtask, the RR-TNM dataset was split into training, validation, and test sets in the ratio of 12:6:9. The dataset was split on a case basis so that radiology reports diagnosing the same case did not appear in multiple splits. For each report of the training and validation sets, T, N, and M factors are available for participants of the RR-TNM subtask. For each report of the training/validation/test sets, Japanese reports are available, which can be used as input to systems of the TNM staging.

3.2 Overview of our system

Input of our system is a radiology report written in Japanese, and output of our system is values of the TNM staging estimated from the radiology report. For extracting and classifying the TNM staging, three components were mainly used in our system: ChatGPT, LangChain, and regular expression. Each radiology report was processed using the three components sequentially. Figure 1 shows the schematic illustration of the process using ChatGPT and LangChain in this study. After the process shown in Figure 1, the output of our system was formatted as "| T:3,N:2,M:1 |". Finally, a regular expression was used to extract the values of T, N, and M factors.

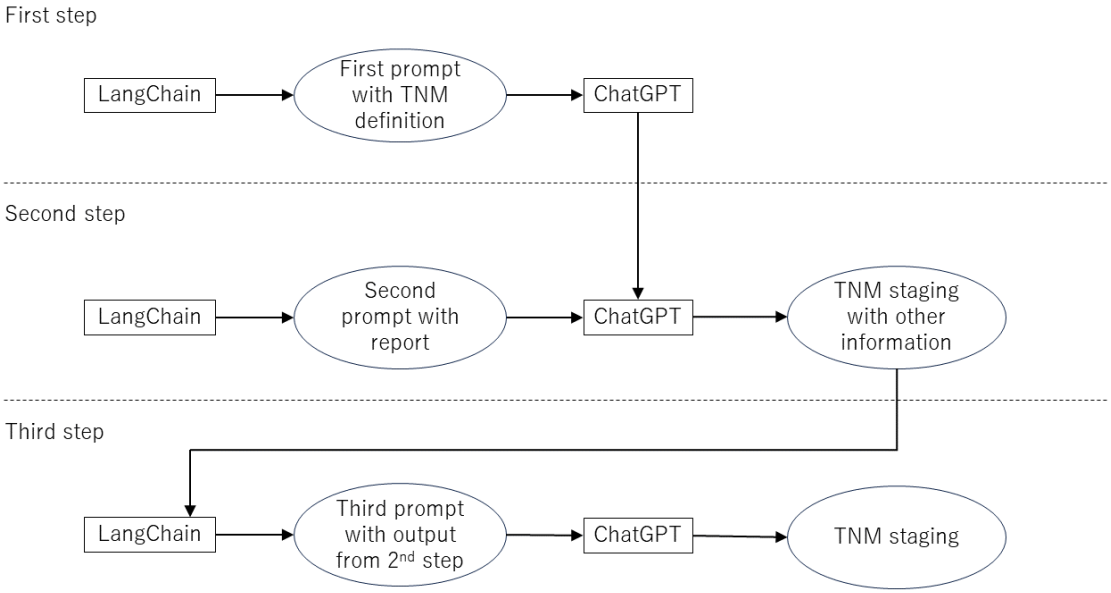

Figure 1. Overview of the process using ChatGPT and LangChain. When using ChatGPT and LangChain, the process of extracting and classifying the TNM staging from radiology reports was divided into 3 steps in our system. In the first step, custom prompts with definition of the TNM staging were provided with ChatGPT using LangChain. In this step, the prompts help ChatGPT to understand knowledge about TNM staging and focus on the TNM staging in its responses. In the second step, the prompt for extracting and classifying the TNM staging from radiology reports was created using LangChain. Here, the report was given to ChatGPT. After the second step, ChatGPT outputs the TNM staging estimated from the report. However, the output frequently included not only TNM staging but also other information. Therefore, as the third step, the third prompt with the output of the second step was given to ChatGPT for extracting only TNM staging. After the third step, the output was formatted as "| T:3,N:2,M:1 |".

3.3 ChatGPT and LangChain

In our system, ChatGPT was used via OpenAI's API. In the API calling, LangChain was used to control ChatGPT. Specifically, we used ConversationChain and ConversationBufferMemory of LangChain when inputting data to ChatGPT via the API. The following parameters were used for ChatGPT: temperature = 0 and top_p = 1.

As shown in Figure 1, the process of extracting and classifying the TNM staging from radiology reports is divided into three steps in our system. In the first step, custom prompts with definition of the TNM staging are provided with ChatGPT using LangChain. In this step, the prompts help ChatGPT to understand knowledge about the TNM staging and to focus on the TNM staging in its responses. In the first step, the following prompt was given to ChatGPT using LangChain. By using this custom prompt in the first step, ChatGPT retained the memory of the TNM definition in the second step. LungChain was useful for retaining the memory.

あなたは熟練の呼吸器外科医です。
以下の TNM の定義を元にして後に与えられる読影レポートの TNM 分類を作成してください。
TNM の定義:
When the following criteria of T factor are used for lung cancer staging, please describe the T factor of the reports.
the criteria:
The T factor is divided into several categories, which are denoted by different subscripts:
TX: The primary tumor cannot be evaluated.
T0: There is no evidence of a primary tumor.
T1: The tumor is 3 cm or smaller and is located in the lung.
T2: The tumor is larger than 3 cm or has spread to nearby structures such as the chest wall, diaphragm, or lining of the lung.
T3: The tumor has invaded the chest wall, diaphragm, or mediastinal pleura (the lining of the space between the lungs).
T4: The tumor has invaded nearby structures such as the heart, great vessels (aorta, pulmonary artery, or superior vena cava), trachea, esophagus, vertebral body, or carina.
the reports:
1) Lung cancer is located at the right upper lobe. The size of the cancer is 4.5 cm.
2) There is a lung nodule in the left lower lobe, which is suspected to a lung cancer. The suspected lung cancer invades left PA.
According to the given criteria, the T factor of the reports are:
Lung cancer is located at the right upper lobe, with a size of 4.5 cm. Based on the criteria, the T factor for this report would be T2 since the tumor is larger than 3 cm.
There is a lung nodule in the left lower lobe, which is suspected to be lung cancer and invades left PA (pulmonary artery). Based on the criteria, the T factor for this report would be T4 since the tumor has invaded a nearby structure, the pulmonary artery.
所見に記載が無い場合には異常な所見が無かったものと考えてください。

Next, the second prompt for extracting and classifying the TNM staging from radiology reports was created using LangChain. Here, a radiology report was given to ChatGPT with the second prompt. The prompt in the second step was as follows.

以下の読影レポートを理解して TNM 分類を出力してください。
当てはまるものがない場合には空にしてください。
出力は正確に次の形式に従ってください：
{T:value,N:value,M:value}。
余分な情報や詳細な説明は一切必要ありません。直接 TNM 分類の値だけを出力してください
Report: {report}"

Here, sentences of the Japanese report were inserted into "{report}" of the second prompt for each report. After the second step, ChatGPT would output the TNM staging estimated from the report. However, the output frequently included not only TNM staging but also other information (while we instructed ChatGPT not to output any extra text!). Therefore, as the third step, the third prompt with the output of the second step was given to ChatGPT for extracting only TNM staging. The following prompt was used in the third step.

以下の読影レポートを理解して TNM 分類を出力してください。
当てはまるものがない場合には空にしてください。
出力は正確に次の形式に従ってください： |
T:value,N:value,M:value |
余分な情報や詳細な説明は一切必要ありません。直接 TNM 分類の値だけを出力してください
Report:{output}

Here, the sentences from the output of the second step were inserted into "{output}" of the third prompt. After the third step, the output of ChatGPT was formatted as "| T:3,N:2,M:1 |". Hereafter, the functionality of the third step is referred to as formatting.

3.4 Regular expression

After the process with ChatGPT and LangChain, the output was processed with a regular expression to extract values of T, N, and M factors. The following regular expression was utilized with re module of python: r" T(\S+) *, *N *(\S+) *, *M *(\S+) *". If no match was found for this regular expression, then "error" was used as an estimated value instead of a number. This regular expression matched even if T, N, or M was followed by a non-numeric value. In this case, the non-numeric value was used as the estimated value. In other words, our system could output non-numeric values as T, N, and M factors.

3.5 Postprocessing

Since the organizers' system did not accept non-numeric values for the TNM staging, we performed a postprocessing step on the non-numeric values for our system's results of the test set. In this postprocessing step, the non-numeric values of the T, N, and M factors were replaced with 0.

3.6 Evaluation metrics

The organizers employed four types of accuracies to evaluate systems for the TNM staging based on the labels in the test set. The three types of accuracies were independently calculated for the T, N, and M factors. Additionally, the joint accuracy was determined as the fraction of reports in which the T, N, and M factors were all predicted correctly.

3.7 Implementation options in our system

In our system, there were several implementation options available. These options were as follows:

- Version of ChatGPT: GPT version 3.5 (gpt3.5-turbo) or GPT version 4.
- Use of the TNM definition in the prompt: with and without TNM definition.
- Language of the prompt: English or non-English.
- Use of formatting outputs with ChatGPT (the third step described in Section 3.3): with and without formatting.

To assess the effect of these options, we explored 10 combinations of the four options within our system. For this purpose, the training and validation sets were utilized. Notably, the postprocessing was excluded from this experiment. For the "English" option, Japanese sentences in the prompt were translated into English using machine translation.

4 RESULTS

4.1 Accuracies calculated by the organizers

The four types of accuracies were calculated by the organizers for the test set. The results are shown in Figure 2. The T, N, and M accuracies of our system were 0.3951, 0.8025, and 0.9259, respectively. The joint accuracy of our system was 0.2716.

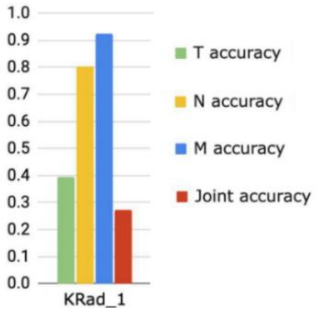

Figure 2. Accuracies of T, N, and M factors and joint accuracy of our system for the test set.

4.2 Implementation options and its effect

We evaluated the effect of the four implementation options in our system. This evaluation was done not to select the best set of implementation options, but to assess the effect of the options. For the 10 combinations of the four options, we calculated the four types of accuracies using the training and validation sets. Figures 3 and 4 show the accuracies of the 10 combinations for the traininig and validation sets, respectively. The combination of the four options used for the test set is shown on the leftmost side of Figures 3 and 4. When evaluating the effect of the implementation options, the cost (in dollars) of the OpenAI's API in GPT version 4 was more than 20 times that of GPT version 3.5.

4.3 Source code of our system

The source code of our system using ChatGPT, LangChain, and the regular expression is shown at the end of this paper.

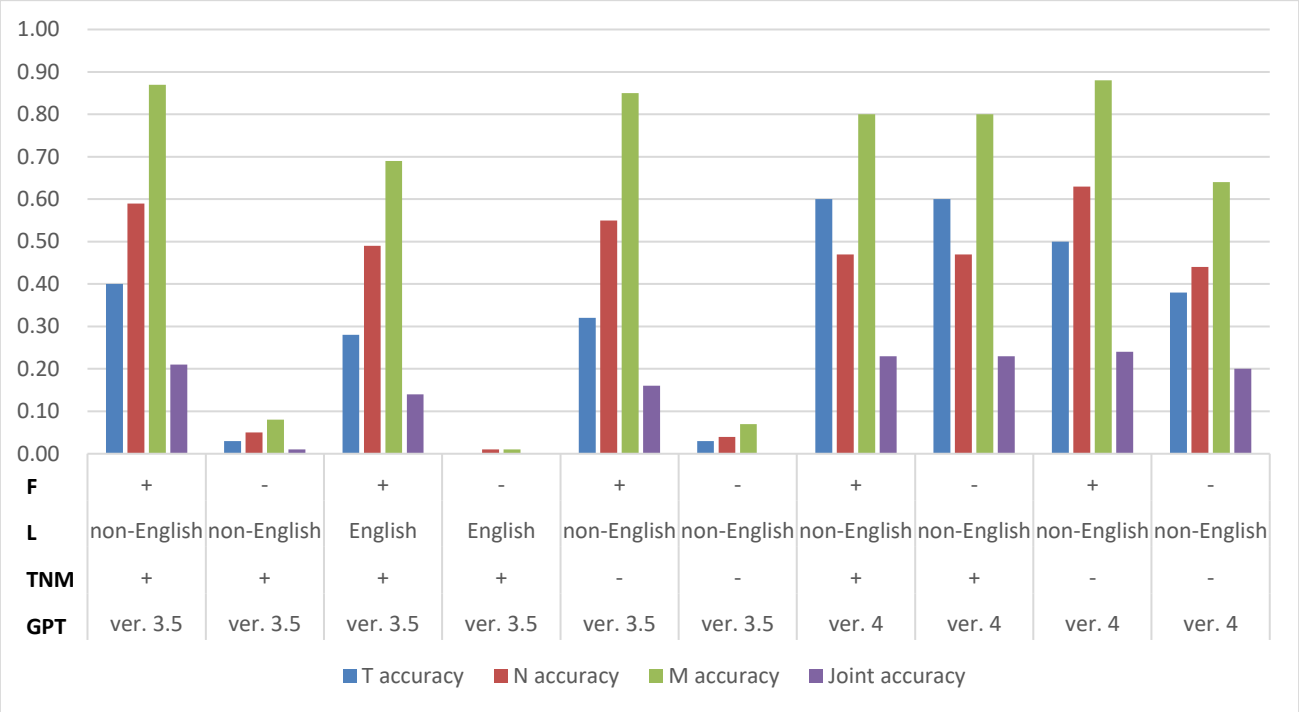

Figure 3. Effect of implementation options in the traininig set. Figure 3 shows the four types of accuracies for the 10 combinations of the implementation options. Abbreviations: F, formatting; L, language; TNM, TNM definition; GPT, version of ChatGPT.

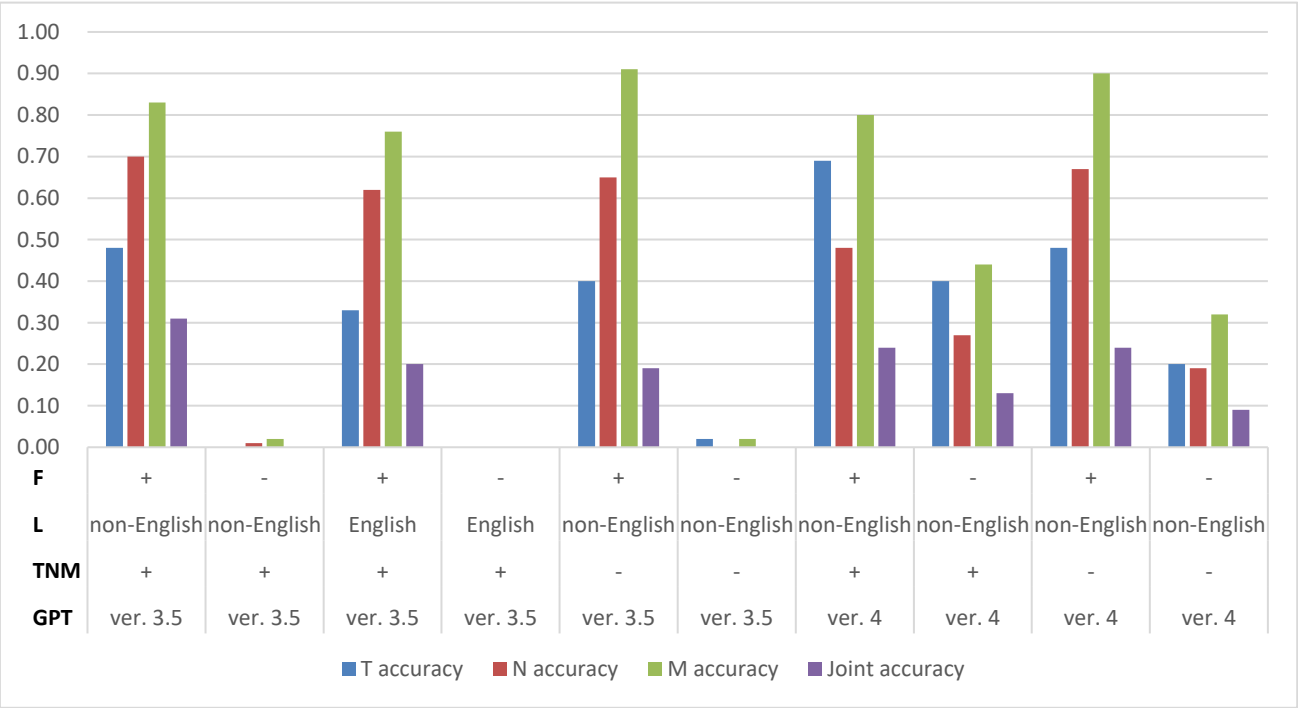

Figure 4. Effect of implementation options in the validation set. Figure 4 shows the four types of accuracies for the 10 combinations of the implementation options. Abbreviations: F, formatting; L, language; TNM, TNM definition; GPT, version of ChatGPT.

5 DISCUSSION

The N and M accuracies of our system for the test set were the highest among the seven submissions in the RR-TNM subtask. This might be attributed to the high performance of ChatGPT in zero-shot classification. However, our system employed the postprocessing, which could potentially affect our accuracies in the test set. Since a binary classification was conducted for the M factor, the influence of the postprocessing might be significant for the M factor. Particularly, if there is an imbalance in the M factor of the RR-TNM dataset, our system's performance might be overestimated. On the contrary, the accuracy of the T factor in our submission was lower than that in the other submissions. According to Figures 3 and 4, if GPT version 4 were used in ChatGPT instead of GPT version 3.5, the accuracy of the T factor in our system could potentially be improved for the test set.

Due to the cost of OpenAI's API, we used GPT version 3.5 in the test set. When comparing GPT version 3.5 with GPT version 4, there was a cost difference of over 20 times. In the present study, we prioritized cost-performance. Therefore, GPT version 4 was not used for estimating the TNM staging for the test set. As shown in Figures 3 and 4, the implementation options used for the prediction in the test set were not the best in the training and validation sets. In situations where performance is critical even with an increase in cost, the use of GPT version 4 might be preferable.

With GPT version 3.5, the responses generated by ChatGPT were inconsistent with our instructions in terms of the formatting. On the other hand, with GPT version 4, there was less deviation from our prompt instructions. This is another aspect where ChatGPT with GPT version 4 is considered to hold an advantage.

As shown in Figures 3 and 4, the formatting with ChatGPT was useful. However, the effectiveness is different between version 3.5 and version 4. In the present study, the non-English prompt was better than the English prompt in Figures 3 and 4. This may be due to the fact that Japanese radiology reports were used in the RR-TNM subtask. The effect of these implementation options should be evaluated in other tasks and datasets in a future study.

6 CONCLUSION

In this manuscript, we have described our approach for the RR-TNM subtask in MedNLP-SC shared task. Our system, utilizing ChatGPT and LangChain, was developed for zero-shot classification of the TNM staging. Our results show that zero-shot classification with ChatGPT may be promising.

ACKNOWLEDGEMENTS

This work was supported by JSPS KAKENHI (Grant Numbers: 22K07665, 23K07154, 23K17229, and 23KK0148).

Declaration of Generative AI and AI-assisted technologies

During the preparation of this work, the authors used ChatGPT in order to analyze the radiology reports of the RR-TNM dataset and prepare this paper. After using this tool/service, the authors reviewed and edited the content as needed and takes full responsibility for the content of the publication.

Contribution

HM performed most of the experiments described in this paper. MN wrote the first draft of the manuscript. All authors participated in several discussions and exchanged ideas. All authors read the paper and approved it.

REFERENCES

1. Nakamura Y, Hanaoka S, Yada S, Wakamiya S, Aramaki E. NTCIR-17 MedNLP-SC Radiology Report Subtask Overview: Dataset and Solutions for Automated Lung Cancer Staging. In Proceedings of the 17th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR-17 National Institute of Informatics (NII). 2023.
2. Siegel RL, Miller KD, Fuchs HE, Jemal A. Cancer statistics, 2022. CA Cancer J Clin. 2022;72: 7–33. doi:10.3322/caac.21708
3. Brierley J.D., Gospodarowicz M.K., Wittekind C. TNM Classification of Malignant Tumours, 8th edition. Union for International Cancer Control. 2016; 1–272.
4. Lim W, Ridge CA, Nicholson AG, Mirsadraee S. The 8th lung cancer TNM classification and clinical staging system: review of the changes and clinical implications. Quant Imaging Med Surg. 2018;8: 709–718. doi:10.21037/QIMS.2018.08.02
5. Fujimoto K, Nishio M, Sugiyama O, Ichikawa K, Cornelius J, Lithgow-Serrano O, et al. Approach for Named Entity Recognition and Case Identification Implemented by ZuKyo-JA Sub-team at the NTCIR-16 Real-MedNLP Task Approach for Named Entity Recognition and Case Identification Implemented by ZuKyo-JA Sub-team at the NTCIR-16 Real-MedNLP Task TEAM NAME. Proceedings of the 16th NTCIR Conference on Evaluation of Information Access Technologies. 2022. pp. 322–329. doi:10.5167/uzh-220151
6. Shuntaro Y, Yuta N, Shoko W, Eiji A. Real-MedNLP: Overview of REAL document-based MEDical Natural Language Processing Task. In Proceedings of the 17th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR-16 National Institute of Informatics (NII). 2022.
7. Devlin J, Chang M-W, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics; 2019. pp. 4171–4186. doi:10.18653/v1/N19-1423
8. Matsuo H, Nishio M, Nogami M, Zeng F, Kurimoto T, Kaushik S, et al. Unsupervised-learning-based method for chest MRI–CT transformation using structure constrained unsupervised generative attention networks. Sci Rep. 2022;12: 11090. doi:10.1038/s41598-022-14677-x
9. Yamashita R, Nishio M, Do RKG, Togashi K. Convolutional neural networks: an overview and application in radiology. Insights Imaging. 2018;9: 611–629. doi:10.1007/s13244-018-0639-9
10. Casey A, Davidson E, Poon M, Dong H, Duma D, Grivas A, et al. A systematic review of natural language processing applied to radiology reports. BMC Medical Informatics and Decision Making 2021 21:1. 2021;21: 179. doi:10.1186/S12911-021-01533-7
11. Hu D, Zhang H, Li S, Wang Y, Wu N, Lu X. Automatic Extraction of Lung Cancer Staging Information From Computed Tomography Reports: Deep Learning Approach. JMIR Med Inform. 2021;9: e27955. doi:10.2196/27955
12. Gupta K, Thammasudjarit R, Thakkinstian A. NLP Automation to Read Radiological Reports to Detect the Stage of Cancer Among Lung Cancer Patients. Proceedings of the 2019 Workshop on Widening NLP. Florence, Italy: Association for Computational

Linguistics; 2019. pp. 138–141. Available: https://aclanthology.org/W19-3643
13. Aalabdulsalam AK, Garvin JH, Redd A, Carter ME, Sweeny C, Meystre SM. Automated Extraction and Classification of Cancer Stage Mentions from Unstructured Text Fields in a Central Cancer Registry. AMIA Joint Summits on Translational Science proceedings AMIA Joint Summits on Translational Science. 2018. pp. 16–25.
14. Radford A, Narasimhan K, Salimans T, Sutskever I. Improving Language Understanding by Generative Pre-Training. 2018.
15. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language Models are Unsupervised Multitask Learners. 2019.
16. Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, et al. Language Models are Few-Shot Learners. ArXiv. 2020;abs/2005.14165. Available: https://arxiv.org/abs/2005.14165
17. ChatGPT. [cited 10 Aug 2023]. Available: https://chat.openai.com/
18. Liu P, Yuan W, Fu J, Jiang Z, Hayashi H, Neubig G. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. ArXiv. 2021;abs/2107.13586. Available: https://arxiv.org/abs/2107.13586
19. LangChain. [cited 10 Aug 2023]. Available: https://www.langchain.com/

SOURCE CODE

```
import os
import logging
import sys
import pandas as pd
import pickle
import re
from langchain.llms import OpenAI
from langchain.chains import LLMChain
from langchain.prompts import PromptTemplate
from langchain.chains import ConversationChain
from langchain.memory import ConversationBufferMemory


def get_csv(input_path):
    df = pd.read_csv(input_path)
    input_dic = []
    for i in range(len(df)):
        tmp_dic = {}
        tmp_dic["ID"] = df.loc[i, "ID"]
        tmp_dic["T"] = df.loc[i, "T"]
        tmp_dic["N"] = df.loc[i, "N"]
        tmp_dic["M"] = df.loc[i, "M"]
        with open("../../dataset/" + str(df.loc[i, "ID"]) + ".txt", "r") as f:
            tmp_str = f.read()
            tmp_dic["Report"] = tmp_str
        if i < 5:
            input_dic.append(tmp_dic)
        else:
            input_dic.append(tmp_dic)
    return input_dic

def process_tnm(process_dic, prefix=""):
    os.environ["OPENAI_API_KEY"] = ""
    logging.basicConfig(stream=sys.stdout, level=logging.DEBUG, force=True)
    llm = OpenAI(temperature=0, model_name="gpt-3.5-turbo")
    memory = ConversationBufferMemory()
    conversation = ConversationChain(llm=llm, memory=memory)
    # TNM 定義ファイルの読み込み
    with open("../../init_data/tnm_en.txt", "r") as f:
        docu_text = f.read()
    first_input = (
        "あなたは熟練の呼吸器外科医です。\
            以下の TNM の定義を元にして後に与えられる読影レポートの
TNM 分類を作成してください。\
            TNM の定義:"
        + docu_text
        + "\n 所見に記載が無い場合には異常な所見が無かったものと考えて
ください。"
    )
    conversation.predict(input=first_input)
    first_pickel = pickle.dumps(conversation.memory)
    # TNM classification prompt
    classification_prompt = PromptTemplate(
        input_variables=["report"],
        template="以下の読影レポートを理解して TNM 分類を出力してくださ
い。\
            当てはまるものがない場合には空にしてください。\
            出力は正確に次の形式に従ってください：| T:value,N:value,M:value
|\
                余分な情報や詳細な説明は一切必要ありません。直接 TNM 分類
の値だけを出力してください\n\
                Report:{output}",
    )
    classification_chain = LLMChain(llm=llm, prompt=classification_prompt)
    result_list = []
    for tmp_dic in process_dic:
        conv = ConversationChain(llm=llm, memory=pickle.loads(first_pickel))
        input_str = (
            "以下の読影レポートを理解して TNM 分類を出力してください。\
            当てはまるものがない場合には空にしてください。\
            出力は正確に次の形式に従ってください：
{T:value,N:value,M:value}。\
                余分な情報や詳細な説明は一切必要ありません。直接 TNM 分類
の値だけを出力してください\n\
                Report:"
            + tmp_dic["Report"]
        )
        result_str_1 = conv.predict(input=input_str)
        result_str_2 = classification_chain.run(result_str_1)
        tmp_dic_result = tmp_dic.copy()
        tmp_dic_result["result"] = result_str_1
        tmp_dic_result["classification"] = result_str_2
        result_list.append(tmp_dic_result)
        pattern = r" T(\S+) *, *N *(\S+) *, *M *(\S+) * "
        tmp_dic_result["error_str"] = ""
        match = re.search(pattern, result_str_2)
        try:
            label_t = match.group(1)
        except:
            label_t = "error"
            tmp_dic_result["error_str"] = result_str_2
        try:
            label_n = match.group(2)
        except:
            label_n = "error"
            tmp_dic_result["error_str"] = result_str_2
        try:
            label_m = match.group(3)
        except:
            label_m = "error"
            tmp_dic_result["error_str"] = result_str_2
        tmp_dic_result[f"{prefix}_t"] = label_t
        tmp_dic_result[f"{prefix}_n"] = label_n
        tmp_dic_result[f"{prefix}_m"] = label_m
        print(label_t, label_n, label_m)
    return result_list

if __name__ == "__main__":
    input_dic = get_csv("../../../dataset/test/sample_submission.csv")
    result = process_tnm(input_dic, "gpt-3.5-turbo")
    df_result = pd.DataFrame(result)
    df_result.to_excel("result.xlsx")
```