Master Thesis

Extracting Activity Information with LLMs Using GPT-Generated Data

Chuqiao Guo

a thesis submitted in partial fulfilment of the requirements for the degree of

MA Linguistics
(Text Mining)

Vrije Universiteit Amsterdam

Computational Lexicology and Terminology Lab
Department of Language and Communication
Faculty of Humanities

Supervised by: Piek Vossen
2nd reader: Pia Sommerauer

Submitted: June 30, 2024

Abstract

The increasing aging population has led to a rise in the number of chronic patients, thereby causing a shortage of healthcare facilities. This project aims to develop an NLP system that can extract patients' daily activities from their conversations with a healthcare chatbot. In this way, doctors can monitor the patients' conditions remotely, therefore alleviating the burden on medical resources. To construct the dataset for the experiment, a zero-shot approach was employed to prompt GPT-3.5-turbo to generate natural conversational data describing patients' daily activities, categorised under the International Classification of Functioning, Disability, and Health (ICF) framework. GPT-4o was then prompted to label the extensive training dataset. An experiment was conducted to compare the performance of three distinct NLP systems: a rule-based model, a fine-tuned BERT model, and a GPT-prompting system. The results demonstrated that the GPT-prompting system achieved the best overall performance with an F1-score of 0.70. The fine-tuned BERT model showed moderate performance with an F1-score of 0.62, while the rule-based model proved unsuitable for processing dynamic conversational data, achieving an F1-score of 0.30.

Declaration of Authorship

I, Chuqiao Guo, declare that this thesis, titled *Extracting Activity Information with LLMs Using GPT-Generated Data* and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a degree degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: Jun 30, 2024

Signed:

Acknowledgments

First and foremost, I would like to thank my advisor, Piek Vossen, for his invaluable guidance, patience, and insightful feedback throughout this research. I also extend my gratitude to my second reader, Pia Sommerauer, for dedicating her time to provide a critical analysis of this work. Also, I am deeply grateful to the healthcare professionals in the A-PROOF team—Sabina, Edwin, and Marike—for their collaboration and valuable insights on my project.

Special thanks to all the staff and colleagues in the CLTL who have worked with me throughout this year. Your support has been instrumental in helping me explore the realm of NLP.

Lastly, I extend my heartfelt thanks to my family and friends for their unwavering support and encouragement throughout my academic journey. Their belief in me has been a constant source of motivation.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

The Netherlands is experiencing a demographic shift with an ageing population (Statistics Netherlands (CBS), 2024). Recent studies indicate that among those over 75, only 7% are without chronic illnesses, while 80% have multiple conditions (Agnieszka et al., 2021). This trend will likely worsen as the elderly population increases. Chronic diseases require regular monitoring, significantly burdening healthcare resources. Traditional monitoring methods, involving frequent hospital visits, strain the system heavily. In response to this challenge, the development of a healthcare chatbot is proposed. This chatbot would communicate regularly with patients, record their utterances, and organise key information from them. Additionally, it will analyse this data to document it into the patient's medical dossier. By leveraging such a system, doctors can monitor the health conditions of chronic disease patients remotely, thus alleviate the burden on medical resources.

This project precisely represents the initial phase in the extensive development process of the healthcare chatbot, focusing on designing an NLP system capable of detecting activity-related information within patient-chatbot interactions.

1.1 Goal and research questions

This exploratory project aims to develop an NLP system that can effectively detect the information of patients' daily activities through conversations held with a chatbot. For example, in the sentence "I read books last Friday in the garden with Alex", the detected information is expected to be:

- *Event*: read books
- *Place*: in the garden
- *Time*: last Friday
- *Participant*: I, Alex

Achieving this research goal involves the process of utilising prompt engineering with GPT to generate patient-chatbot conversations and corresponding activity-related labels, designing a rule-based model as a baseline, fine-tuning a pre-trained BERT model, and conducting error analysis across different systems.

The research question of this project is to examine the feasibility of designing an NLP system to detect activity-related information within patient-chatbot conversations. This primary question can be further divided into the following sub-questions:

- Which activities should be detected from the ICF categories?
- How to prompt ChatGPT to generate natural conversations and corresponding semantic role label annotation?
- Which NLP system performs the best in detecting activity information (Event, Participant, Time, and Place)?

The first sub-question will be discussed qualitatively in Chapter 3, Section 3.1.2, with cooperation of healthcare experts from the A-PROOF team. The second sub-question will be elaborated in Chapter 3, Section 3.1.3 and 3.1.4. The third sub-question will be further discussed in Chapter 6.

1.2 Outline

The thesis is outlined as follows: Chapter 2 provides literature review on the past and current research of large natural language generation approaches, and prompt engineering techniques. Chapter 3 elaborates on the detailed methodology for generating conversational data and labels, the distribution of the datasets, the experimental setup, and the evaluation criteria. Chapter 4 reports the preliminary results of the experiment, after validating the three systems on the development dataset. Chapter 5 reveals the optimisation process for all the NLP systems. Chapter 6 further reports the system performance on the test set using optimal models. Chapter 7 analyses the typical errors across all three systems. Chapter 8 discusses the limitations of this project, along with potential improvements and new directions for future research. Chapter 9 draws a conclusion.

Chapter 2

Related Work

This chapter will present the key concepts and introduction of the method used in this project. Section 2.1 presents the structure and usage of the International Classification of Functioning, Disability and Health (ICF) framework. The category and definition of various human activities provided by ICF will be used to guide the generation of the text dataset. In addition, Section 2.2 introduces the techniques of natural language generation, which will be further used to generate the conversational data, and 2.3 elaborates on the techniques of text classification, including the GPT-prompting technique, which will be used to generate labels for the training set." Furthermore, Section 2.4 explains the the mechanism of Semantic Role Labelling (SRL) task, which is also the inspiration of the current project.

2.1 The ICF Framework

The International Classification of Functioning, Disability and Health (ICF) is a framework developed by the World Health Organization (WHO) that provides a standardised approach to understanding and measuring health and disability levels (Organization, 2001). The ICF framework categorises consists of two parts, *ICF Category* and *ICF Qualifier*. The *ICF Qualifiers* quantifies human functioning on a universal scale, enabling a standard assessment of the severity and extent of health conditions and disabilities. On the other hand, *ICF Category* provides the definition across four distinct domains: body functions, activity and participation, environmental factors, and body structures. The activity and participation domain further categorises the activities of daily living (ADL) into nine specific categories (Figure 2.1). This framework encompasses essential activities such as self-care, mobility, and communication, providing a comprehensive approach to categorising an individual's functional status. Recently studies have shown its effectiveness in the healthcare sector, for determining the impact of chronic diseases on patients' everyday lives (Wolff et al., 2002), assessing the daily functional capabilities of elderly individuals (Guralnik et al., 1994), and evaluating the rehabilitation outcomes of stroke patients (Salter et al., 2005).

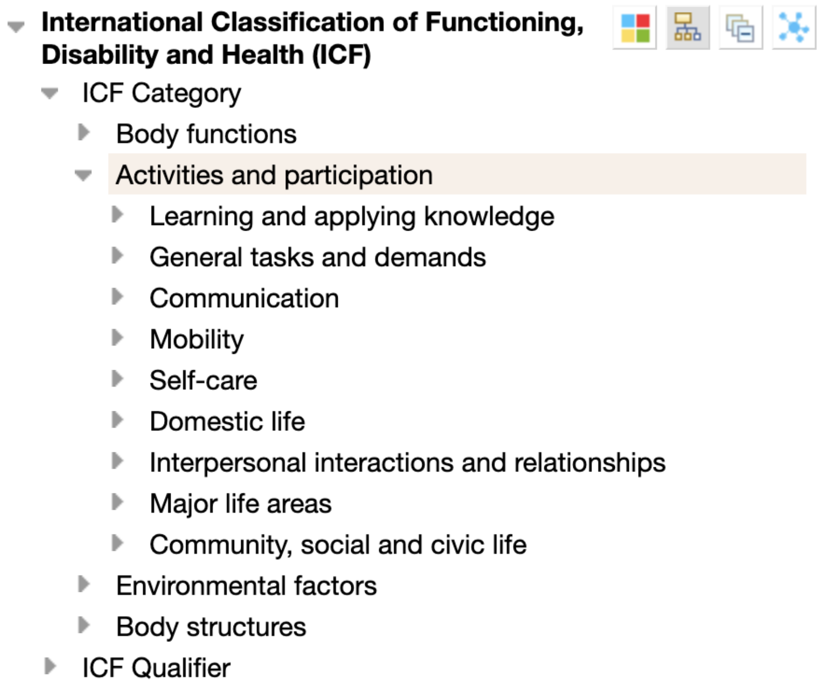

Figure 2.1: ICF Hierarchy: Activities and Participation

2.2 Natural Language Generation

2.2.1 Rule-based Approach

Rule-based approaches in the field of Natural Language Generation use pre-defined templates that are generated following certain linguistic rules. Reiter and Dale (1997) presented a systematic approach to the rule-based NLG process, which consists of three stages: content determination, sentence planning, and linguistic realisation. Rule-based NLG techniques are widely adopted in contexts that are simple and fixed, such as weather reports, restaurant ordering systems, and promotional emails. However, these approaches often struggle in flexible and complex scenarios, as they require extensive manual effort to create and maintain the rule sets, making them less suitable for dynamic or highly variable domains (Deemter et al., 2005; Gatt and Krahmer, 2018).

2.2.2 Deep Learning Approach

In contrast to rule-based methods, deep learning approaches leverage vast training datasets and neural network architectures to learn linguistic patterns. The emergence of recurrent neural networks (RNNs) and transformers has greatly enhanced the capabilities of deep learning NLG systems. The ability to generate natural and versatile text of the deep learning models has been proven across various applications, such as summarisation and machine translation (Liu and Lapata, 2019; Vaswani et al., 2023). However, the training process requires vast computational resources (Strubell et al., 2020) and may exhibit bias from the training dataset (Bender et al., 2021).

In November 2022, OpenAI [1] released the Generative Pre-trained Transformer

[1] OPENAI:https://openai.com

(GPT) series, and revolutionised the field. These models employ self-attention mechanisms to manage long-range dependencies and produce high-quality text (Vaswani et al., 2023; Radford et al., 2019). By pre-training on large corpora and fine-tuning for specific tasks, they can generate high-quality text that is both coherent and contextually relevant (Devlin et al., 2018; Liu et al., 2019).

Prompt engineering is the technique that controls the GPT series models' behaviour by specifically crafted input prompts. This technique is proven to be effective and efficient across various NLP tasks (Brown et al., 2020). Compared to other transformer-based models, prompt engineering allows for quick and directional adjustments, thereby enhancing the model's interpretability.

2.3 Text Classification

This section provides an overview of various text classification methods. Section 2.3.1 covers traditional machine learning approaches, while Section 2.3.2 elaborates on the advanced and versatile techniques utilising pre-trained Large Language Models (LLMs).

2.3.1 Traditional Machine Learning

Traditional machine learning methods, including Naive Bayes and Support Vector Machines (SVM) algorithms, have been foundational in the field of text classification (McCallum et al., 1998; Joachims, 1998). These methods typically involve representing text data through features and applying various classifiers to classify the text data into pre-defined categories (Sebastiani, 2002). For example, the SVM, which is one of the most well-known machine learning models, seeks to find the optimal hyperplane that maximises the margin between different classes, therefore separating the classes with the maximum possible margin. In this way, the SVMs can be used to generalise effectively to unseen data (Joachims, 1998)

2.3.2 Leverage Pre-trained Large Language Models

Fine-tune BERT

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained transformer model that captures contextual relationships between words in a bidirectional manner. This mechanism allows it to understand the nuances of language more effectively (Devlin et al., 2018). The fine-tuning process of BERT involves taking a pre-trained BERT model and further training it on a specifically labelled dataset for a particular task. During this process, the weights of the pre-trained model are adjusted to achieve optimal performance for the new task. This approach allows the model to leverage the linguistic knowledge acquired from the pre-trained dataset while being adapted to output results for the target task. This approach has proven to be highly effective, as BERT's deep bidirectional understanding of language allows it to capture complex patterns and dependencies in text.

Prompt GPT

Prompting GPT (Generative Pre-trained Transformer) represents a paradigm shift in how text classification tasks can be approached. Unlike traditional fine-tuning, prompt-based methods involve designing specific prompts that direct the pre-trained model to generate the desired output. GPT models, such as the latest version of GPT-4o, are pre-trained on vast datasets and demonstrate robust language generation capabilities. By designing directional prompts, these models can be accommodated to do text classification tasks in zero-shot or few-shot learning settings, which require little or no task-specific training data.

For example, with proper prompts, the model can successfully perform a sentiment classification task, where the model is required to predict "positive" or "negative" labels for a given text (Bu et al., 2023). It generates the corresponding outputs based on the pre-trained understanding, and can effectively reduce labour costs for labelling the dataset and computing resources for fine-tuning. Research has demonstrated that by prompting GPT models properly, they can achieve competitive performance on a variety of text classification tasks, showcasing their ability to understand and generate relevant and coherent text with minimal task-specific data (Brown et al., 2020).

2.4 Semantic Role Labelling

The methodology for extracting activity-related information in this project is inspired by the principles of semantic role labelling (SRL) tasks. Specifically, it draws upon the PropBank-style SRL framework, where the main verb of a sentence functions as the predicate, and its associated arguments form a frameset that provides a structured representation of the semantic relationships (Johansson and Nugues, 2008). In the context of this project, the predicate is analogous to the event being mentioned, while the arguments, such as agent, patient, and instrument, correspond to the event details including the time, place, and participants.

By leveraging this SRL-based approach, we ensure a systematic extraction of activity-related information from patient-chatbot conversations. This framework allows for the precise identification of key elements within the dialogues, facilitating a deeper understanding of the patient's daily activities and interactions. The structured representation not only enhances the accuracy of information extraction but also supports the subsequent analysis and interpretation of the data, ultimately contributing to more effective healthcare interventions and resource allocation.

In dependency-based SRL tasks, predicates and their corresponding arguments are first identified and classified, followed by determining their relationships (Gildea and Jurafsky, 2002), (Surdeanu et al., 2008). For example, if there are two predicates in one sentence, this method accurately maps each argument to its respective predicate, ensuring that the semantic meaning is correctly conveyed.

This project focuses solely on the identification and classification steps; specifically, it involves detecting event-related tokens from a given conversation and assigning properties such as place, time, and participants to these events.

Chapter 3

Methodology

In this chapter, Section 3.1 explains the generation process of the dataset used in the research, and Section 3.2 presents the overview statistics of the dataset. In addition, Section 3.3 explains the set-up of the main experiment, consisting of the three NLP systems in extracting activity information. Finally, Section 3.4 illustrates the optimisation and evaluation process of the three systems.

Figure 3.1 presents the flowchart of the project.

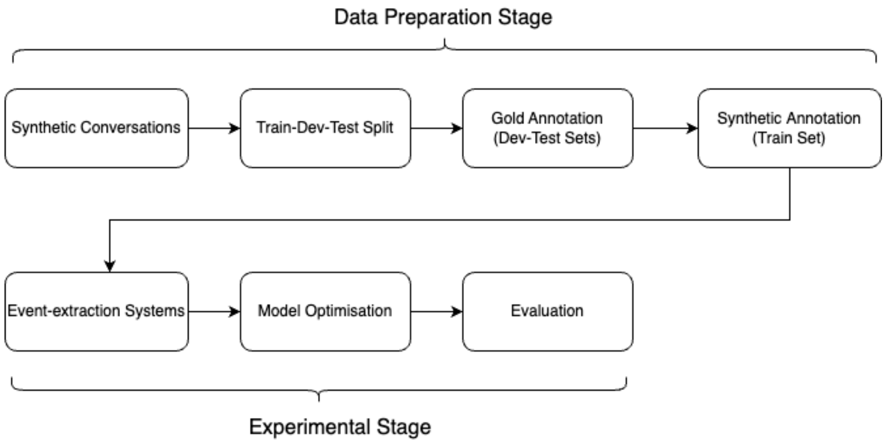

Figure 3.1: Flow Chart of the Project

3.1 Data

Due to the privacy protection of medical data, the user conversation is inaccessible. Therefore, the text data used in this paper were generated through prompt engineering using GPT-3.5-turbo and subsequently labelled by GPT-4o. When generating the conversational data, the events defined by ICF framework was applied as guidance, which ensures a comprehensive representation of various events in patients' daily lives.

3.1.1 Overview of the ICF

The International Classification of Functioning, Disability and Health (ICF) (Organization, 2001) is a comprehensive framework developed by the World Health Organization

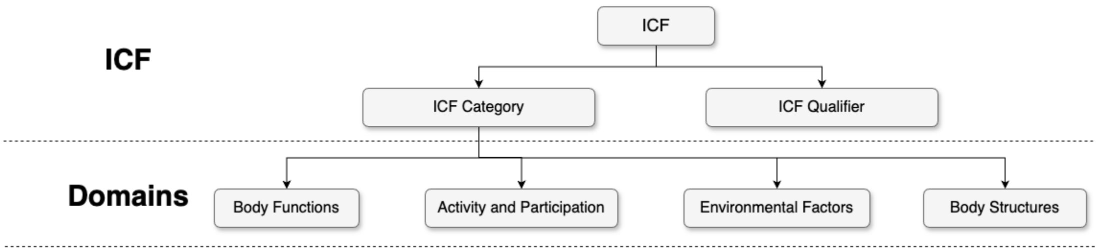

Figure 3.2: ICF - Four Domains

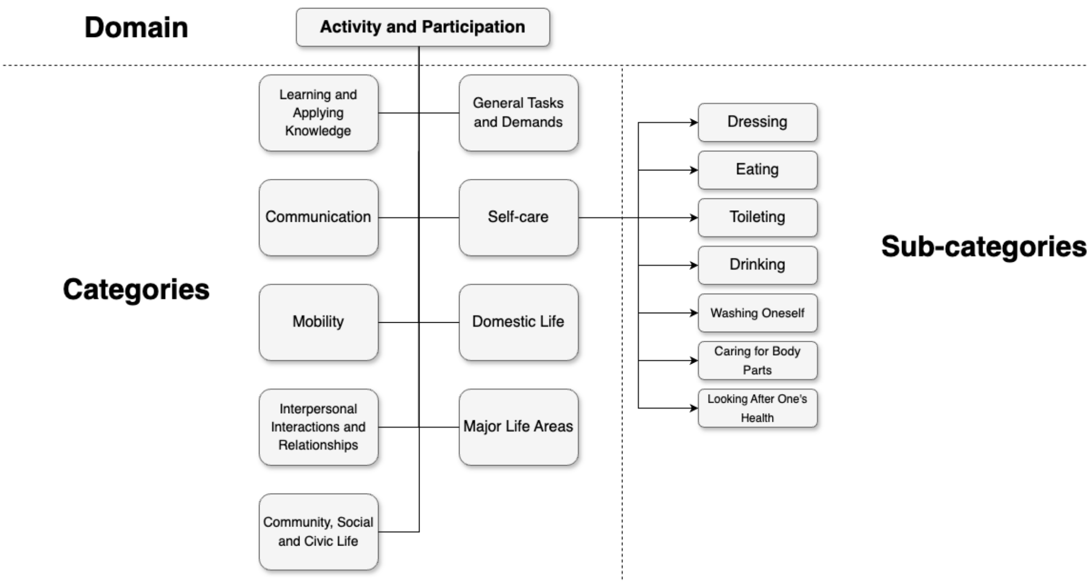

Figure 3.3: ICF Categories and Sub-categories

(WHO) for measuring health and disability levels. It establishes standardised definitions and measurements for the field of healthcare within various contexts.

The ICF framework (Figure 3.2) mainly consists of two parts: *ICF Category* and *ICF Qualifier*. *ICF Category* categorises functioning and disability into four main domains: (1) Body Functions, (2) Activities and Participation, (3) Environmental Factors, and (4) Body Structures. *ICF Qualifier* is a universal scale designed to quantify human functioning.

3.1.2 Selection of Category

Within *ICF Category*, each domain is further divided into various categories. The domain focusing on describing the engagement in activities: "Activities and Participation", includes nine categories: *Learning and Applying Knowledge, General Tasks and Demands, Communication, Mobility, Self-care, Domestic Life, Interpersonal Interactions & Relationships, Major Life Areas, and Community, Social & Civic Life*. Each of them is further explained in sub-categories. The information provided in the sub-categories involves specific activities, such as "Dressing", "Eating", and "Toileting" in the sub-category "Self-care" (Figure 3.3).

Despite the comprehensiveness of these nine categories, not all are suitable for guiding the generation of conversations in the current task. In extracting activity information from a given conversation, the objective is to detect specific events rather than ambiguous ones. For example, the category *Interpersonal Interactions & Relationships* is ambiguous due to its broad and varied nature. This category might encompass a wide range of interactions, making it challenging to pinpoint precise events when generating conversations. Consequently, we prioritise categories with more concrete definitions to ensure the quality of generated conversations.

The final selection of categories included in the dataset is made in consultation with healthcare experts from the A-PROOF team, who possess extensive knowledge of both the ICF framework and real user scenarios. The topics under the following three categories are used to guide the generation of the conversations, as they are concrete and unambiguous.

Mobility is about moving by changing body position or location or by transferring from one place to another, by carrying, moving or manipulating objects, by walking, running or climbing, and by using various forms of transportation.

Self-care is about caring for oneself, washing and drying oneself, caring for one's body and body parts, dressing, eating and drinking, and looking after one's health.

Domestic Life is about carrying out domestic and everyday actions and tasks. Areas of domestic life include caring for one's belongings and space, acquiring food, clothing and other necessities, household cleaning and repairing, caring for personal and other household objects, and assisting others

3.1.3 Conversation Generation

The generated conversations are intended to accurately reflect real user scenarios, including detailed accounts of patients' daily activities along with contextual information such as time, location, and participants. To ensure these conversations' comprehensiveness and topical diversity, the system was prompted with the definitions of each subcategory under the three selected categories (Figure 3.4).

Each conversation is prompted to contain no more than 10 utterances, and each line should have no more than 20 tokens. The topic of the conversation should fall under the subcategories of the ICF categories. The three conversations from three distinct subcategories below are examples of GPT-generated conversations.

(1) Mobility
Chatbot: *How have you been feeling lately? Have you been managing daily tasks well?*
Patient: *Well, I've had some trouble getting out of my chair to get into bed at night.*
Chatbot: *I see. When does that usually happen?*
Patient: *It usually happens when my back is acting up and I feel stiff.*
Chatbot: *Is there anything that makes it easier for you to transition?*
Patient: *Yes, using a grab bar next to the bed helps me to pull myself up.*
Chatbot: *That's great to know! How about other daily activities, are there any other occasions where mobility is challenging?*
Patient: *Yes, standing for long periods makes my legs feel weak and I need to sit down often.*

```
query = [
    {"role": "system", "content": "Think as you are a real human."},
    {"role": "system", "content": """Generate one small and natural
    conversation without greetings by playing the roles of a friend (F)
    and an elderly patient (P). The conversation has 6 utterances,
    mentioning one or two events. Each utterance should be completed
    and has less than 20 tokens. The format is as below:
    F: utterance
    P: utternace
    F: utterance
    ...
    """},

    {"role": "user", "content": f"The topic of the conversation is
    about {category} events. In terms of functioning, {category} is
    about {definition}. {category} events include {events}."},

    {"role": "user", "content": f"The patient can talk with the
    friend about his/her daily life which can reflect his/her
    functioning in {category}. The friend should ask when and
    how the event occurs. "}
    ]
```

Figure 3.4: Prompt template for conversation generation

(2) Domestic Life
Chatbot: *How was your day today?*
Patient: *I spent the morning cleaning the house.*
Chatbot: *Do you clean the house every day?*
Patient: *No, I have a cleaning schedule for each day of the week.*
Chatbot: *That's quite organized. What do you do on other days?*
Patient: *On Tuesdays, I focus on doing the laundry and tending to the plants.*
Chatbot: *Sounds like a productive routine. How do you manage everything?*
Patient: *I've found that having a schedule helps me stay on track and manage my tasks efficiently.*
Chatbot: *That's great to hear. It must make things a lot easier for you.*

(3) Self-care
Chatbot: *How often do you usually do your skincare routine?*
Patient: *Oh, I do it every morning and night before bed.*
Chatbot: *That's great! How about brushing your teeth, when do you do that?*
Patient: *I brush my teeth after every meal and before bed.*
Chatbot: *Good habit. Do you find cutting your nails challenging?*
Patient: *Yes, it's a bit harder for me now. I do it every other week.*
Chatbot: *I see. What about caring for your general health? How do you manage that?*
Patient: *I take my medication daily and go for regular check-ups at the clinic.*

Chatbot: *That's important. Taking care of yourself is key.*

To examine the quality of the generated texts, we invited the experts in the A-PROOF team to evaluate and provide insights into the generated data. Overall, the synthetic data are fairly natural, despite the instances within *domestic life* category exhibiting slight discrepancies when compared to real-life conversations.

The generated dataset (Table 3.1) comprises 940 conversations in total (280, 420, and 240 for Mobility, Self-care, and Domestic Life respectively). Upon manual inspection, it is observed that several sentences contain nonsense text, particularly at the end of the conversations. This issue arises due to the model's context drift and memory constraints (Radford et al., 2019), which cause it to lose track of the conversation's context over extended interactions. As a result, the model may generate sequences of text that lack coherence or meaningful content.

A semi-automated data-cleaning method is utilised to eliminate conversations containing gibberish words, specifically filtering out those with excessively long tokens (exceeding 20 letters) and sentences (surpassing 20 words). After data cleaning, the dataset consists of 795 conversations and 6914 utterances.

	Raw Data		**Cleaned Data**	
Category	**Conversation**	**Utterance**	**Conversation**	**Utterance**
Mobility	280	2765	241	2049
Self-care	420	4312	335	2944
Domestic Life	240	2413	219	1921
Total	940	9490	795	6914

Table 3.1: Dataset by Category

Due to the manual labelling required for the development and test sets, and considering constraints in both time and resources for this project, I have designated 7% (54 conversations) of the dataset for the test set, another 7% (54 conversations) for the development set, and set the remaining 86% (687 conversations) as the training set. The split of the dataset into these subsets was performed randomly.

3.1.4 Label Generation

The activity information within each conversation, including event, participant, time, and place, is extracted and labelled for the training dataset. These labels are formatted using the BIO (Beginning, Inside, Outside) system: B-event, I-event, B-participant, I-participant, B-time, I-time, B-place, I-place, and O. Due to the large size of the training dataset and the limited timeframe of this project, manually annotating the training dataset is challenging.

Research shows that GPT-3 can generate high-quality labels when properly prompted due to its advanced natural language understanding and generation capabilities. GPT-4o, released in May 2024, is reported to be more competent than GPT-3.5-turbo, and faster and more cost-effective than GPT-4. Therefore, prompt engineering techniques using GPT-4o are applied to label the training dataset. This approach leverages the strengths of GPT-4o to automate the annotation process, ensuring consistency and accuracy in the labels while significantly reducing the time and effort required compared to manual annotation.

Large language models like GPT-4o have been trained on extensive datasets and possess a broad understanding of natural language. To leverage its pre-trained capabilities, the zero-shot method is employed to generate labels for activity information. In the zero-shot method, no specific examples are provided in the given prompt. It maximises the versatility of GPT-4o, which is beneficial for performing such a new task without designated answers.

Prompts for labels generation

First, an initial prompt is applied to generate activity information comprising event, participant, time, and place, along with their respective sentence and token identifiers (Figure 3.6). The system processes the conversation in segments, with each segment provided as input one at a time as strings (Figure 3.5). Each segment results in an output formatted as a JSON object, as illustrated below:

```
[
{'activity_index': 1,
'activity': 'activity 1',
'activity_sentence_id': 1,
'activity_token_ids': [1, 2],
'participants': 'participant 1',
'participants_sentence_id': 1,
'participants_token_ids': [3],
'place': 'place 1',
'place_sentence_id': 2,
'place_token_ids': [5],
'time': 'time 1',
'time_sentence_id': 3,
'time_token_ids': [7]},
{
...
}
]
```

```
1. How have your days been lately, managing different tasks and activities?
2. Oh, it has been quite busy even with just the household chores.
3. When do you usually find time to tackle those domestic tasks?
4. I reserve some time in the mornings and evenings to get them done.
5. How do you balance your time between chores and personal activities?
6. Making to-do lists and following a routine definitely helps in managing it all.
7. Have you assigned particular chores for each day of the week?
8. Yes, I find it easier doing specific tasks on different days for better organization.
9. Do you enjoy checking those chores off the list once they're completed?
10. Absolutely, keeps me focused and motivated to

<class 'str'>
```

Figure 3.5: Initial conversational data input

```
query = [
    {"role": "system", "content": "You are an expert in healthcare. I will provide you \
    some conversations between a chatbot and an elderly person. Please extract the information \
    contained in each conversation: 'activity index', 'activity', 'participants', 'place', \
    and 'time', and format every activity as a list."},
    {"role": "system", "content": "When extracting the information, please use the words \
    and phrases appeared in the original conversation. Suppose sentence id is reset when \
    starting a new conversation, and token id is reset when starting a new sentence. \
    Please also indicate the sentence id and token id of the activity information."},
    {"role": "system", "content": f"The conversation index for this conversation is \
    {conversation_index}. If the conversation contains more than one activity, \
    generate a list for each activity, using the same conversation index but different\
     activity indices. For a new activity, increment the activity index by 1."},
    {"role": "system", "content": "Generate the activity information only based on\
     the conversation. Do not use any external information."},
    {"role": "system", "content": "If there are no participants, place, or time \
    of the activity mentioned in the conversation, please mark as 'None' in the output."},
    {"role": "user", "content": f"Conversation: {con}"},
    {"role": "system", "content": "Please provide the output in the following JSON format: \
    [{'activity_index': 1, 'activity': 'activity 1', 'activity_sentence_id': 1, \
    'activity_token_ids': [1, 2], 'participants': 'participant 1', 'participants_sentence_id': 1, \
    'participants_token_ids': [3], 'place': 'place 1', 'place_sentence_id': 2, \
    'place_token_ids': [5], 'time': 'time 1', 'time_sentence_id': 3, 'time_token_ids': [7]}, {...}]. \
    Please provide the output without Markdown code blocks, and do not include the newline marker \\n in the output."},
]
```

Figure 3.6: Initial prompt template for generating activity information

In this prompt, GPT-4o is expected to generate sentence ID and token ID based on specific rules, in order to ensure unique identification of tokens within each conversation. The sentence ID starts at 1 and increments by 1 with each new sentence. Similarly, the token ID starts at 1 for each sentence and increments by 1 for each token within that sentence.

Upon manual inspection, the generated activity information is found to be generally accurate. However, GPT-4o exhibits shortcomings in precisely indicating the position of certain tokens within the conversation, despite the provision of rules for counting sentence and token identifiers. For example, in the sentence ”I usually get up very early in the morning”, the token ID for the temporal phrase ”in the morning” should be [7, 8, 9], whereas GPT-4o may output [5, 6, 7].

This limitation arises mainly because large language models such as GPT-4o are designed for linguistic tasks rather than explicit mathematical computations. Their transformer architecture is optimised for predicting the next token in a sequence, based on the vast amount of pre-training data (Vaswani et al., 2017). Therefore, while GPT-4o demonstrates proficiency in extracting activity information, the architecture emphasises contextual predictions over numerical precision, which makes it challenging to accurately pinpoint the token's position within sentence boundaries. Also, in the tokenization process, the numerical information may be further split into sub-tokens, resulting in disruption of the model's mathematical ability (Sennrich et al., 2016). Therefore, if positional information are pre-calculated and provided alongside the conversation input, GPT-4o may improve its ability to output accurate token indices.

To efficiently locate the activity information extracted by GPT-4o and map them back to the conversational data, the format of the input conversation is improved. In the improved version (Figure 3.7), I use a tokenized database as the input in the prompt given to the system. This new database provides the pre-calculated conversation ID, sentence ID, and token ID for each token within a conversation, and inputs each conversation with this information one at a time to the system as strings. By pre-providing this accurate information, the deficiencies in GPT's counting are expected to be mitigated, thereby better assisting GPT-4o in accurately locating activity information within the conversation and sentences.

```
Conversation 1: 1      1    1    How    - 1    1    2    have   - 1    1    3    your
Conversation 2: 2      1    1    How    - 2    1    2    has    - 2    1    3    your
Conversation 3: 3      1    1    How    - 3    1    2    was    - 3    1    3    your
Conversation 4: 4      1    1    How    - 4    1    2    's     - 4    1    3    your
Conversation 5: 5      1    1    What   - 5    1    2    do     - 5    1    3    you
Conversation 6: 6      1    1    How    - 6    1    2    do     - 6    1    3    you
Conversation 7: 7      1    1    So     - 7    1    2    ,      - 7    1    3    how
Conversation 8: 8      1    1    How    - 8    1    2    has    - 8    1    3    your
Conversation 9: 9      1    1    How    - 9    1    2    was    - 9    1    3    your
Conversation 10: 10    1    1    How    - 10   1    2    was    - 10   1    3    your
Conversation 11: 11    1    1    How    - 11   1    2    have   - 11   1    3    you
Conversation 12: 12    1    1    How    - 12   1    2    has    - 12   1    3    your
Conversation 13: 13    1    1    How    - 13   1    2    are    - 13   1    3    you
Conversation 14: 14    1    1    How    - 14   1    2    's     - 14   1    3    your
Conversation 15: 15    1    1    How    - 15   1    2    was    - 15   1    3    your
Conversation 16: 16    1    1    How    - 16   1    2    was    - 16   1    3    your
Conversation 17: 17    1    1    Do     - 17   1    2    you    - 17   1    3    enjoy
Conversation 18: 18    1    1    How    - 18   1    2    are    - 18   1    3    you
Conversation 19: 19    1    1    How    - 19   1    2    's     - 19   1    3    your
Conversation 20: 20    1    1    How    - 20   1    2    was    - 20   1    3    your
```

Figure 3.7: Improved conversational data input

```
query = [
    {"role": "system", "content": "You are an expert in healthcare. I will provide you some conversations between\
     a chatbot and an elderly person. Please extract the information contained in each conversation: \
    'activity index', 'activity', 'participants', 'place', and 'time', and format every activity as a list."},
    {"role": "system", "content": "Each token in the conversation is provided with three information:\
     conversation id, sentence id, and token id."},
    {"role": "system", "content": "When extracting the information, please use the words and phrases appeared \
    in the original conversation. Please also indicate the sentence id and token id of the activity information."},
    {"role": "system", "content": f"The conversation index for this conversation is {conversation_index}. \
    If the conversation contains more than one activity, generate a list for each activity, using the\
     same conversation index but different activity indices. For a new activity, increment the activity index by 1."},
    {"role": "system", "content": "Generate the activity information only based on the conversation.\
     Do not use any external information."},
    {"role": "system", "content": "If there are no participants, place, or time of the activity mentioned\
     in the conversation, please mark as 'None' in the output."},
    {"role": "user", "content": f"Conversation: {con}"},
    {"role": "system", "content": "Please provide the output in the following JSON format:\
     [{'activity_index': 1, 'activity': 'activity 1', 'activity_sentence_id': 1, 'activity_token_ids': [1, 2], \
    'participants': 'participant 1', 'participants_sentence_id': 1, 'participants_token_ids': [3], \
    'place': 'place 1', 'place_sentence_id': 2, 'place_token_ids': [5], 'time': 'time 1', 'time_sentence_id': 3, \
    'time_token_ids': [7]}, {...}]. Please provide the output without Markdown code blocks, \
    and do not include the newline marker \\n in the output."},
]
```

Figure 3.8: Improved prompt template for generating activity information

Upon manual inspection of ten conversational turns, the improved version—incorporating conversation ID, sentence ID, token ID, and token—demonstrates perfect accuracy in identifying the token's position within the conversation, as evidenced by the correct output of the corresponding sentence ID and token ID. Additionally, to ensure that tokenizing the input conversation does not compromise accuracy, specific elements such as event, time, place, and participant were also scrutinised. The examination confirms that the extracted activity information remains consistent, whether the conversation is input in a tokenized form or as a whole.

3.1.5 Annotation of Development and Test Set

To investigate the performance of the three NLP systems in extracting activity information, the development and test data are labelled manually by the author. Ideally, annotation by multiple annotators would enhance reliability and enable the calculation of the Inter-Annotator Agreement (IAA); however, such an approach is not feasible within the time constraints of this thesis.

The annotation follows these guidelines:

- Label the event and corresponding place, time, and participant within each conversation
- Label only the events that have already happened, and set aside the planned events
- Leave the annotation blank if the place, time, or participant is missing.

3.1.6 Event-Argument Relations

In standard Semantic Role Labeling (SRL) tasks, the relationship between a sentence's predicate and its arguments is determined to better illustrate the sentence's semantic meaning. This project, as discussed in Section 2.4, focuses solely on the detection and classification phases of SRL, rather than the full range of tasks. This limited scope can introduce ambiguity when multiple distinct events are mentioned within a single conversation, particularly if tokens are shared across these events (e.g., scenarios where two events share the same time, place, and participants). However, semi-automatic inspection using a script to detect such scenarios indicates that this ambiguity is rare, except in cases where the participant is a default, such as "I".

Another source of ambiguity arises when multiple events are mentioned within the same conversation. Manual inspection of the 108 conversations in the development and test datasets reveals that this is common. Nonetheless, the event-argument relationships can be effectively identified using dependency relations, as arguments are typically presented within complete sentences rather than isolated words. Moreover, the generated data retains the information about event-argument relationships, providing a foundation for future research aimed at generating event-related information and event-argument relations.

3.2 Label Distribution

The training dataset consists of 5,965 labelled utterances from 687 conversations. Figure 3.9 shows the distribution of the labels in the training set. Generally, the label "O" dominates the training set with a proportion of 89.7%, as most of the tokens do not belong to any event.

To facilitate a better understanding of the label distribution for event-related tokens, the label *O* was excluded from the analysis. As illustrated in Figure 3.10, the categories "event" and "time" dominate the label distribution. Specifically, *I-event* constitutes 44.5%, *I-time* for 25.1%, *B-event* comprises 13.9%, and *B-time* represents 9.6%. In contrast, the proportions for "participant" and "place" are relatively smaller, with *B-participant* at 2.4%, *I-place* at 2.2%, *B-place* at 1.7%, and *I-participants* at 0.5%.

Figure 3.9: Training Set - Label Distribution

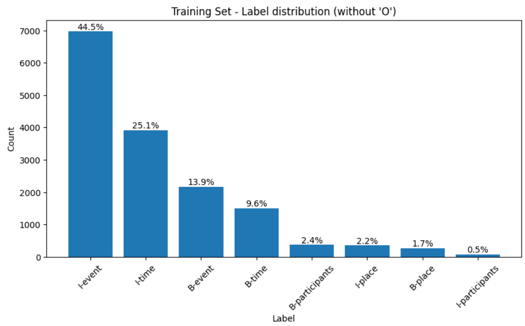

Figure 3.10: Training Set - Label Distribution (without "O")

The labels in the development set show a comparable distribution (Figure 3.11, 3.12). The development set includes 54 conversations and 468 sentences. The label *O* dominates the dataset with a proportion of 76.5%, which is 13.2% lower than that of the GPT-labelled training set. After excluding the label *O*, the labels in the categories "event" and "time" are the most prevalent. Notably, *B-participant* represents 10.5% of the event-related tokens, which is higher than that in the training set (2.4%). This discrepancy arises because the training set was labelled using prompt engineering with GPT-4o due to the time constraints of this project, while the development set was annotated manually. The GPT-prompting method is found to be ineffective in accurately capturing participant tokens, which will be further discussed in later chapters.

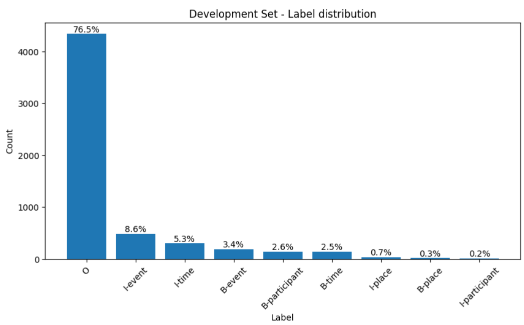

Figure 3.11: Development Set - Label Distribution

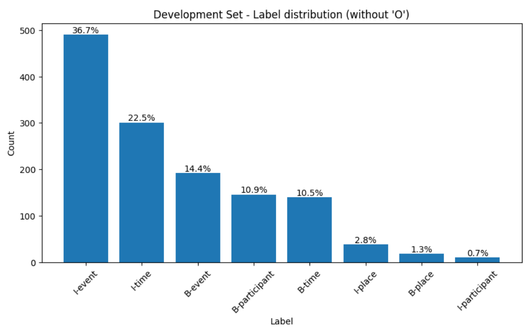

Figure 3.12: Development Set - Label Distribution (without "O")

Similar to the development set, the manually labelled test dataset comprises 478 utterances from 54 conversations, which were generated using GPT-3.5-turbo. The label distribution of the test set, as shown in Figure 3.13 and 3.14, closely mirrors that of the development set. The non-event label *O* constitutes 77.5% of the test set. Excluding these *O* tokens, the most prevalent labels are "event" and "time", with *I-event* making up 39.0%, *B-event* 13.8%, *I-time* 20.7%, and *B-time* 9.7%. Additionally, the label *B-participant* accounts for 11.5%, closely matching the development set's 10.9%.

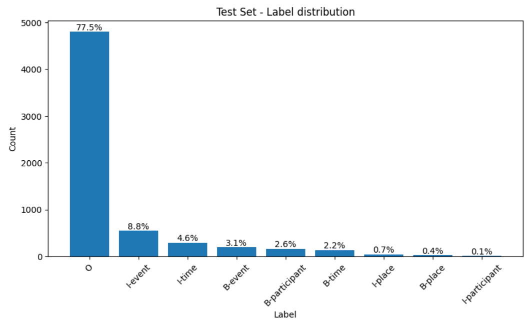

Figure 3.13: Test Set - Label Distribution

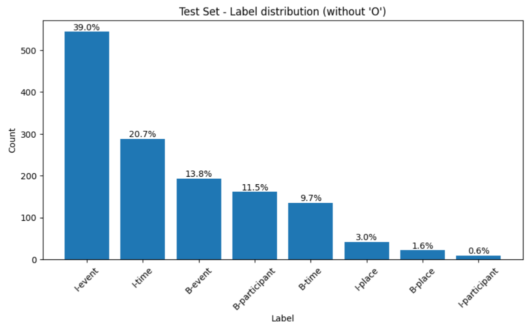

Figure 3.14: Test Set - Label Distribution (without "O")

3.3 Experiment Setup

The experiment encompasses three distinct NLP systems. The first system, described in Section 3.3.1, serves as the baseline. It is a rule-mapping system that utilises spaCy's dependency parsing and Named Entity Recognition (NER) processor to generate universal rules for extracting activity information. The second system, detailed in Section 3.3.2, employs a GPT-generated training dataset to fine-tune the pre-trained multilingual BERT model. This system aims to leverage the contextual understanding capabilities of the BERT model for enhanced performance. The third system, discussed in Section 3.3.3, examines the zero-shot labeling capability of GPT-4o by comparing the activity labels produced by GPT-4o with the gold labels.

3.3.1 Rule-based System

This system detects the activity information using spaCy's dependency parser and the Named Entity Recognition (NER) processor. SpaCy (Honnibal and Montani, 2017) is an efficient open-source library for NLP, which provides a wide range of tools and processors for different NLP tasks.

Dependency Parser

Dependency parsing is a technique that illustrates the syntactic relations between tokens within a sentence . In syntactic parsing, the structure of a sentence is modelled as a tree, where the root node represents the predicate of the sentence, and the branches are the dependents. In this hierarchical syntactic tree, each token serves as a node, with the syntactic relationships connecting them (Nivre et al., 2016). When parsing a certain token, the token which governs its syntactic role is recognised as its "ancestor", and the dependent token is called its "child", and each dependent of it can further have dependents. Thus, a nested structure that captures the complexity of the syntactic relation is created.

For example, in the sentence "I wash my face every morning and before bedtime" (Figure 3.15), the verb "wash" is the predicate, also the head of the clause, and it governs the syntactic roles of other words in the sentence. "I" is a child of "wash" with a nominal subject (nsubj) relationship, indicating the person who performs the action. Similarly, "face" is a child of "wash" with a direct object (dobj) relationship, indicating the thing that is being washed.

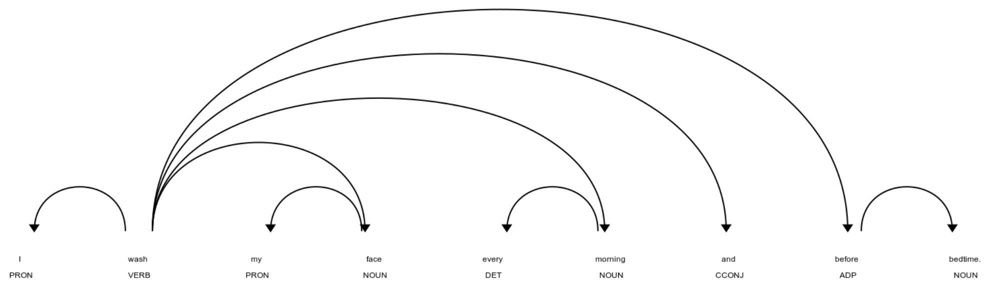

Figure 3.15: Dependency Tree

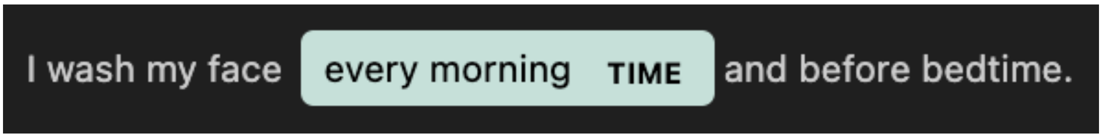

Figure 3.16: Named Entity Recognition

Named Entity Recognition Processor

The NER processor in spaCy is designed to detect named entities in a given text, and classify these entities to predefined categories. These categories includes names of people (PERSON), organizations (ORG), locations (LOC), dates (DATE), times (TIME), and etc.. For example, in the sentence "I wash my face every morning and before bedtime", "every" and "morning" are recognised as "TIME", indicating a temporal reference (Figure 3.16).

In addition, previous studies have demonstrated spaCy's NER processor to be highly effective, exhibiting its high accuracy in various real-world applications (Honnibal and Montani, 2017).

Designed Rules

The rule-based system leverages the ability of dependency parsing as well as the NER processor provided by spaCy to design specific rules to extract activity information.

In this system, each conversation will be split into several sentences to process. The events within the sentence are identified through verb phrases using dependency parsing. First, the verb and its direct object form the core of the verb phrase (VP). To ensure the completeness of the verb phrase, the possessive modifier and determiner associated with the direct object are also included as part of the VP. For example, in the sentence "I wash my face every morning and before bedtime" (3.15), the verb "wash", its direct object "face", and the possessive modifier "my" (modifying "face") together form the verb phrase "wash my face".

Time expressions are recognised if the token is a temporal modifier of the verb, or the token is as recognised as "TIME" or "DATE" by the NER processor. Also, the possessive modifier and determiner associated with the token are included as a part of the time phrase.

Similarly, place expressions are recognised using the NER processor. The token is recognised as place-related if it is marked as "GPE" or "LOC" by the NER processor.

Participants are identified by examining noun chunks within the sentences. Specifically, the focus is on noun chunks where the root dependency is a nominal subject (nsubj) and the head of this root is a verb. This ensures that the noun chunk is acting as the subject of the verb, which often indicates the participant in the action described by the verb. Additionally, named entities recognised as persons (with the entity type PERSON) are also included as participants. To maintain accuracy, certain criteria are applied to exclude noun chunks that are unlikely to represent valid participants. Specifically, noun chunks that contain non-specific pronouns such as 'it', 'that', 'this', 'there', or 'here' are filtered out. For example, in the sentence "I went shopping with Amy", "I" and "Amy" are identified as the participants because "I" is a nominal subject, and the named entity of "Amy" is "PERSON".

After extracting all the activity information, it will be mapped back to the dataset using unique positional identifiers: conversation index, sentence ID, and token ID. The dataset will then be annotated using the BIO labelling system, where the beginning of a span is marked with "B-", the subsequent tokens within the span are marked with "I-", and tokens that do not contain activity information are marked as "O". The quality of the labels generated by the rule-based system will be evaluated by comparison with the manually annotated test set.

3.3.2 Fine-tuning BERT System

The fine-tuned multilingual BERT model will serve as the first experiment group. As mentioned in 2.3.2, this model was pre-trained on a large, labelled multilingual dataset, using tasks such as masked language modelling (MLM) and next sentence prediction (NSP). Through fine-tuning the manually labelled development set, the model can be tailored to annotate the event, time, place, and participants mentioned in a given conversation.

This project fine-tunes the pre-trained model *"bert-base-multilingual-cased"* using a training set annotated by GPT-prompts to assess the impact of fine-tuning on transformer model performance. The fine-tuning process was executed on a GPU with CUDA support in Google Colab to ensure efficient processing. The AdamW optimiser was utilised with a learning rate of 1e-4. Gradient clipping was implemented with a threshold of 1.0 to prevent gradient explosion. The training was conducted over 10 epochs, with each batch consisting of 16 samples.

After the fine-tuning process, the model from the best-performing epoch is selected for optimisation. The hyper-parameters will be tuned and the model will be tested on the manually annotated development set. After tuning, the model will be assessed using the manually annotated test set.

3.3.3 Zero-shot GPT-Prompting System

The zero-shot GPT-prompting system will serve as the second experiment group. As discussed in Section 3.1.4, GPT-4o is capable of extracting activity information using the zero-shot prompting technique. The effectiveness of labels generated via prompts will first be assessed by comparison with a manually generated evaluation set. Afterwards, by analysing the evaluation metrics, the system will be optimised by adjusting the prompts, and further evaluated on the manually annotated test set.

3.4 Evaluation and Optimisation

3.4.1 Model Evaluation

The evaluation of the three NLP systems in detecting activity information from the patient-chatbot conversations will be conducted at the token level. This process will determine how accurately each system predicts labels compared to the gold labels in the test set. To quantify their performance, precision, recall, and F1 scores will be calculated for each system, and the results will be presented using confusion matrices.

Token-level analysis is applied to this study due to the potential length of gold spans in certain contexts, which may encompass multiple words. For example, in the sentence "I watered the plants at 10 AM in the morning," the gold span for the temporal phrase is "10 AM in the morning." Nevertheless, detecting only "10 AM" by the system is deemed acceptable, as it maintains semantic coherence. This is because "10 AM" and "in the morning" are semantically interchangeable, thereby avoiding potential confusion. With span-level analysis, such cases would be marked as true negatives because the detected span overlaps with, but does not exactly match, the gold span. Therefore, the span-level evaluation cannot accurately reflect the system's actual performance.

The performance of each system will be assessed based on key metrics, including precision, recall, and F1-score. Given the prevalence of the label "O," a macro-average approach will be employed for these metrics rather than a weighted average. This ensures a balanced evaluation of the system's overall performance, taking into account each class equally, and provides a more robust assessment in scenarios where class imbalances may impact weighted averages.

In particular, the macro-average F1-score will be regarded as the primary criterion for evaluating the model, as it strikes a balance between precision and recall, making it a suitable overall metric. Precision, considered independently, reveals the extent of errors present in the tokens identified as event-related information. In contrast, the recall score gauges the model's proficiency in correctly recognising event information, calculated by (True Positives) / (True Positives + False Negatives).

After calculating the evaluation metrics, the errors of each label category will be analysed, and patterns of common mistakes will be identified. The error analysis will help in understanding the strengths and weaknesses of each system, providing insights into areas that require improvement. This comprehensive evaluation will guide future enhancements in the NLP systems, and shed light on more accurate detection of event information.

3.4.2 Model Optimisation

The model performance will first be assessed on the development set. Subsequently, the overall performance and errors of each system will be carefully examined. Optimised methods will then be employed: for the rule-based system, rules will be adjusted to encompass more comprehensive scenarios; for the BERT-based model, hyperparameters will be tuned to achieve better performance; and for the GPT-prompting system, prompts will be refined to address output errors. Finally, the optimised systems will be evaluated on the test set, which will form the final results of this experiment.

Chapter 4

Preliminary Results

In this chapter, the three systems are evaluated using the development set. After evaluation, the preliminary results are reported in Section 4.1, 4.2 and 4.3. These results will inform the system optimisation process, which will be further explained in Chapter 5.

4.1 Rule-based System

The performance of the rule-based system, as illustrated in Table 4.1, indicates overall sub-optimal effectiveness, with a macro average precision of 0.25, a macro average recall of 0.29, and a macro average F1-score of 0.23.

Specifically, the category *B-participant* performs relatively well, achieving F1-scores of 0.50. The event categories are moderate, with an F1 score of 0.31 for *B-event* and 0.30 for *I-event*. However, the system struggles significantly with place-related categories (*B-place* and *I-place*), scoring 0.00 in all three metrics, which shows its major issues in correctly identifying place-related tokens.

	Precision	**Recall**	**F1-Score**	**Support**
B-event	0.19	0.88	0.31	192
B-participant	0.34	0.96	0.50	146
B-place	0.00	0.00	0.00	20
B-time	0.22	0.32	0.26	138
I-event	0.29	0.31	0.30	494
I-participant	0.25	0.10	0.14	10
I-place	0.00	0.00	0.00	44
I-time	0.62	0.11	0.19	300
O	0.83	0.68	0.75	4351
Accuracy			0.62	5695
Macro Avg	0.30	0.37	0.27	5695
Weighted Avg	0.71	0.62	0.64	5695

Table 4.1: Classification Report of Rule-based System

4.2 BERT System

After fine-tuning the pre-trained *bert-base-multilingual-cased* model, the performance is assessed on the development set. Table 4.2 illustrates its performance by category.

From Table 4.2, we can observe that the performance of the model is reasonably moderate, with a macro average precision of 0.69, a macro average recall of 0.58, and a macro average F1-score of 0.60.

Specifically, while the model excels in recognising non-event tokens (F1=0.92), it exhibits significant difficulties in predicting places (*B-place*: F1=0.30, *I-place*: F1=0.49). In contrast, the model demonstrates strong performance in predicting temporal tokens, achieving an F1-score of 0.76 for *B-time* and 0.80 for *I-time*. In addition, it tends to overtag *B-participant* (F1=0.17), as evidenced by a high precision of 0.67 coupled with a very low recall of 0.10.

	Precision	**Recall**	**F1-Score**	**Support**
B-event	0.63	0.67	0.65	192
B-participant	0.67	0.10	0.17	146
B-place	0.30	0.30	0.30	20
B-time	0.78	0.75	0.76	138
I-event	0.63	0.73	0.68	494
I-participant	0.50	0.40	0.44	10
I-place	0.53	0.45	0.49	44
I-time	0.78	0.82	0.80	300
O	0.92	0.93	0.92	4351
Accuracy			0.86	5695
Macro Avg	0.64	0.57	0.58	5695
Weighted Avg	0.86	0.86	0.86	5695

Table 4.2: Classification Report of BERT Model

4.3 GPT-Prompting System

The GPT-prompting system outperforms the fine-tuned BERT model slightly, achieving a macro average precision of 0.97, a macro average recall of 0.56, and a macro average F1-score of 0.60 (Table 4.3). Similar to the BERT model, the GPT-prompting system also shows strong performance in predicting temporal tokens, achieving F1-scores of 0.71 for *B-time* and 0.79 for *I-time*. In addition, it is notable that the system performs reasonably well for *I-participant* (F1=0.59), but struggles with *B-participant*, which has a low F1-score of 0.29. Similarly, while the F1-score for *I-place* reaches 0.58, the model performance in predicting *B-place* is also at a low level (F1=0.20).

	Precision	**Recall**	**F1-Score**	**Support**
B-event	0.60	0.55	0.57	192
B-participant	0.64	0.18	0.29	146
B-place	0.29	0.35	0.32	20
B-time	0.78	0.65	0.71	138
I-event	0.59	0.62	0.60	494
I-participant	0.71	0.50	0.59	10
I-place	0.72	0.48	0.58	44
I-time	0.83	0.75	0.79	300
O	0.90	0.93	0.91	4351
Accuracy			0.85	5695
Macro Avg	0.67	0.56	0.60	5695
Weighted Avg	0.84	0.85	0.84	5695

Table 4.3: Classification Report of GPT-Prompting System

Chapter 5

Model Optimisation

5.1 Rule-based System

5.1.1 Methodology

The rule-based system is demonstrated to be ineffective in extracting place-related tokens (F1=0.00). Upon analysing error cases and examining the pre-defined rules, it was observed that the NER processor does not label any tokens as "GPE" or "LOC". This is because these labels are typically assigned to larger geographical entities or significant locations, such as countries, cities, and well-known landmarks. However, in the conversational dataset where patients describe their daily activities, locative phrases are more likely to refer to common places like "in the garden" or "at home".

Therefore, the rules are enhanced to improve the performance in identifying place-related tokens. The updated rules now include a pre-defined list of locative prepositions such as "in", "on", "at", "near", and others. As each sentence is processed, if a token is in the locative preposition list, all of its dependent tokens (children in the dependency tree) are classified as part of a place.

5.1.2 Results

The updated rules successfully lift the performance in recognising place-related tokens, with recall = 0.55, and F1-score = 0.08 for *B-place*, and F1-score = 0.11 for *I-place* (Table 6.2). The recall rate for the category *B-event* stands at 0.86, however, the precision for this category is notably low at 0.18. This low precision indicates that the system is incorrectly labelling many tokens as *B-event* that do not represent locations, which results in a high number of false positives. The reason is that the updated rules are overly broad, leading to a significant overtagging issue.

5.2 Hyper-parameter Tuning for BERT

5.2.1 Methodology

From the confusion matrix (Figure 5.1), it is observed that the model frequently generalises other labels to *O*, resulting in many missed cases. This issue likely arises from the high prevalence of non-event tokens, which prevents the model from adequately learning the nuances of less frequent classes (Huang et al., 2014). To address this, the learning rate was reduced from 1e-4 to 3e-5. A smaller learning rate stabilises the

	Precision	**Recall**	**F1-Score**	**Support**
B-event	0.18	0.86	0.30	192
B-participant	0.34	0.96	0.50	146
B-place	0.04	0.55	0.08	20
B-time	0.22	0.32	0.26	138
I-event	0.29	0.30	0.29	494
I-participant	0.25	0.10	0.14	10
I-place	0.09	0.14	0.11	44
I-time	0.62	0.11	0.19	300
O	0.86	0.65	0.74	4351
Accuracy			0.59	5695
Macro Avg	0.32	0.44	0.29	5695
Weighted Avg	0.74	0.59	0.63	5695

Table 5.1: Classification Report: Optimised Rule-based System

training process, allowing the model to make more precise updates to its parameters (Smith, 2015). This adjustment aims to enhance the model's ability to recognise less frequent classes, thereby increasing its overall robustness and generalisation across all categories (Wilson et al., 2017).

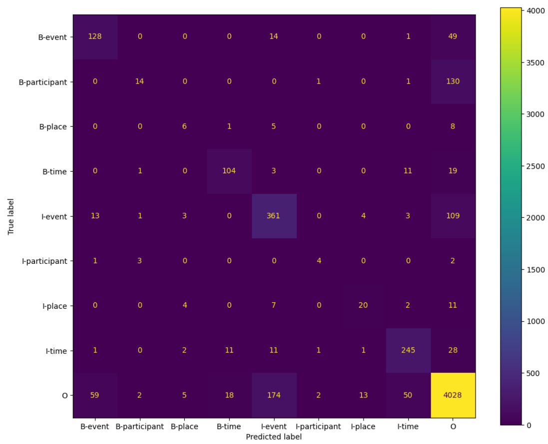

Figure 5.1: Confusion Matrix: Fine-tuned BERT Model

5.2.2 Results

After training with the new learning rate setting, nuanced differences are observed (Table 5.2). The hyperparameter-tuned model surpasses the preliminary in all metrics: it achieves a macro-average precision of 0.68 and a macro-average recall of 0.59. The F1-score, which balances precision and recall, improves noticeably by 3.45% in the hyperparameter-tuned model, reaching 0.60 compared to the original BERT model's 0.58.

Specifically, compared to Table 4.2, the hyperparameter-tuned model (Table 5.3) exhibits significant improvements in recognising *B-event* and *I-event*, with the F1-score increasing from 0.57 to 0.66 and from 0.61 to 0.70 respectively. Similarly, the performance for *I-participant* showed substantial enhancement, with the F1-score rising

Model	Precision	Recall	F1-score
BERT	0.64	0.57	0.58
Tuned BERT	0.68	0.59	0.60
Up	6.25%	3.51%	3.45%

Table 5.2: Model Performance Comparison

from 0.40 to 0.63. Despite these advancements, the model continued to struggle in accurately identifying *B-participant* and *B-place*, achieving F1-scores of 0.11 and 0.15, respectively. Nonetheless, there was a minor improvement in the recognition of *B-participant*, with the F1-score increasing from 0.01 to 0.11.

The comparison underscores the improvements in model performance achieved through learning rate tuning. However, the interpretability of such transformer-based models is relatively limited because its internal multiple layers and attention mechanisms are complex (Vaswani et al., 2017). Therefore, although hyperparameter tuning is effective in optimising the model, the intricate nature of its architecture makes it difficult to directly control the output results through such a tuning process and thus find the optimal model (Shoeybi et al., 2020).

	Precision	Recall	F1-Score	Support
B-event	0.66	0.67	0.66	192
B-participant	0.69	0.06	0.11	146
B-place	0.13	0.17	0.15	18
B-time	0.80	0.74	0.77	140
I-event	0.64	0.77	0.70	490
I-participant	0.67	0.60	0.63	10
I-place	0.52	0.42	0.46	38
I-time	0.77	0.82	0.79	301
O	0.92	0.93	0.92	4338
Accuracy			0.87	5673
Macro Avg	0.64	0.57	0.58	5673
Weighted Avg	0.87	0.87	0.86	5673

Table 5.3: Classification Report: Optimised BERT-based System

5.3 Prompt Engineering for GPT

5.3.1 Methodology

According to the classification report (Table 4.3), the recall of *B-participant* is low (0.11), indicating that the system frequently misses B-participant labels. Upon manual inspection, it was found that the system rarely labels the patient themselves as a participant. In contrast, the gold annotation typically includes the patient as a participant, mostly by labelling the nominal subject. For instance, in the 53rd conversation, 2nd sentence, the gold annotation marks "I" as *B-participant*, while the GPT-prompting system fails to do so. To address this, the new prompt (Figure 5.2) specifies that the patients themselves should also be included in the participant category.

__I__ take the bus to go grocery shopping every Tuesday morning.
(Development Set, Conversation 53, Sentence 2)

In addition, the model struggles more with recognising *B-place* (F1=0.29) than *I-place* (F1=0.50). Upon manual inspection, it was found that the predictions differ from the gold annotation in defining the boundaries of locative phrases. The model tends to drop the preposition preceding the locative noun, whereas the gold annotation includes the preposition and labels the entire prepositional phrase. For example, in the second sentence of the 13rd conversation, the gold annotation for the place is ”in the garden”, whereas the system labels ”the garden” as place.

I wake up early to take care of the flowers ***in the garden***
(Development Set, Conversation 13, Sentence 2)

This discrepancy does not affect the understanding of doctors who read the extracted locative information, as their semantic meaning remains the same with or without the presence of the preceding preposition, but to improve model performance, a refined prompt is employed (Figure 5.2). The new prompt now specifies that the system should include the preposition preceding the noun phrase in the adverbial of place. Consequently, the system is expected to better capture the boundaries of locative phrases.

```
query = [
    {"role": "system", "content": "You are an expert in healthcare. I will provide you some conversations between\
     a chatbot and an elderly person. Please extract the information contained in each conversation: \
    'activity index', 'activity', 'participant', 'place', and 'time', and format every activity as a list."},
    {"role": "system", "content": "Each token in the conversation is provided with three information:\
     conversation id, sentence id, and token id."},
    {"role": "system", "content": "When extracting the information, please use the words and phrases appeared \
    in the original conversation. Please also indicate the sentence id and token id of the activity information."},
    {"role": "system", "content": "When extracting place, please also include the preposition preceding it, if there is any."},
    {"role": "system", "content": "Please indicate the participant as the patient him/herself, if no other participant is mentioned."},
    {"role": "system", "content": f"The conversation index for this conversation is {conversation_index}. \
    If the conversation contains more than one activity, generate a list for each activity, using the\
     same conversation index but different activity indices. For a new activity, increment the activity index by 1."},
    {"role": "system", "content": "Generate the activity information only based on the conversation.\
     Do not use any external information."},
    {"role": "system", "content": "If there are no participants, place, or time of the activity mentioned\
     in the conversation, please mark as 'None' in the output."},
    {"role": "user", "content": f"Conversation: {con}"},
    {"role": "system", "content": "Please provide the output in the following JSON format:\
     [{'activity_index': 1, 'activity': 'activity 1', 'activity_sentence_id': 1, 'activity_token_ids': [1, 2], \
    'participant': 'participant 1', 'participants_sentence_id': 1, 'participants_token_ids': [3], \
    'place': 'place 1', 'place_sentence_id': 2, 'place_token_ids': [5], 'time': 'time 1', 'time_sentence_id': 3, \
    'time_token_ids': [7]}, {...}]. Please provide the output without Markdown code blocks, \
    and do not include the newline marker \\n in the output."},
]
```

Figure 5.2: Refined prompt template for generating activity labels

5.3.2 Results

The refined prompts effectively enhance the system's performance in detecting the default participant (the patients themselves), with the macro average recall for *B-participant* increasing from 0.06 to 0.65 and the macro average F1 score improving from 0.19 to 0.63.

For the category of place, the F1 score for *B-place* rises from 0.29 to 0.44. However, for *I-place*, the precision decreases from 0.68 to 0.34, while the recall increases from

0.39 to 0.71. This suggests that the system is extracting more non-place phrases as places. The addition of a prompt specifically for places likely makes the system more sensitive to place-related terms, resulting in over-tagging, which reduces precision but increases recall.

	Precision	**Recall**	**F1-Score**	**Support**
B-event	0.63	0.60	0.62	192
B-participant	0.61	0.65	0.63	146
B-place	0.32	0.72	0.44	18
B-time	0.73	0.63	0.67	140
I-event	0.59	0.71	0.64	490
I-participant	0.62	0.50	0.56	10
I-place	0.34	0.71	0.46	38
I-time	0.81	0.72	0.76	301
O	0.93	0.90	0.92	4338
Accuracy			0.85	5673
Macro Avg	0.62	0.68	0.63	5673
Weighted Avg	0.86	0.85	0.86	5673

Table 5.4: Classification Report: Optimised GPT-Prompting System

Overall, the initial GPT-prompting system demonstrates strong performance in macro-average precision, achieving a score of 0.69. However, the enhanced model shows significant improvement in macro-average recall, achieving a value of 0.68. This improvement results in a 6.78% increase in the F1-score, rising from 0.59 in the initial model to 0.63 in the enhanced model. The precise adjustments to the prompts can target specific weaknesses in the model's performance more directly, thus providing a more interpretable route to enhancing performance on specific categories.

Model	**Precision**	**Recall**	**F1-score**
GPT-4o	0.69	0.56	0.59
Tuned GPT-4o	0.62	0.68	0.63
Up	-10.14%	21.43%	6.78%

Table 5.5: Model Performance Comparison

Chapter 6

Results

This chapter demonstrates the evaluation process of the three activity detection NLP systems, by testing their performance in the test dataset.

6.1 System Performance

The overall system performances of each model are illustrated in Table 6.1. The GPT-prompting system demonstrated the highest effectiveness, achieving the best scores across precision (0.68), recall (0.73), and F1-score (0.70). In addition, the fine-tuned BERT model performed adequately with a precision of 0.68, recall of 0.62, and an F1-score of 0.62. However, the rule-based system lagged significantly behind the other models, recording the lowest precision (0.30), recall (0.49), and F1-score (0.30), indicating that the traditional method of using dependency parsing and named entity recognition processor to extract certain information is not suitable for conversational data.

Model	**Precision**	**Recall**	**F1-score**
Rule-based	0.30	0.49	0.30
Fine-tuned BERT	0.68	0.62	0.62
GPT-Prompting	0.68	0.73	0.70

Table 6.1: Model Performance Comparison

6.1.1 Rule-based System

Specifically, the rule-based system which is built by pre-designed rules, using spaCy's dependency parser and NER processor, shows varied results across different entity types as illustrated in Table 6.2. The system has moderate performance in identifying *B-participant*, with a precision of 0.31 and an F1-score of 0.46, but struggles significantly with *B-place* and *I-time*, where it achieves F1-scores of 0.09 and 0.10 respectively. Overall, the system has a macro average F1-score of 0.30 and an accuracy of 0.58.

6.1.2 Fine-tuned BERT System

The BERT system employs the pre-trained multilingual BERT model, further fine-tuned on a GPT-4o labelled training dataset. As per the results in Table 6.3, the

	Precision	**Recall**	**F1-Score**	**Support**
B-event	0.16	0.79	0.26	193
B-participant	0.31	0.94	0.46	161
B-place	0.05	0.68	0.09	22
B-time	0.16	0.30	0.21	135
I-event	0.28	0.30	0.29	544
I-participant	0.30	0.33	0.32	9
I-place	0.15	0.36	0.21	42
I-time	0.45	0.06	0.10	288
O	0.87	0.63	0.73	4801
Accuracy			0.58	6195
Macro Avg	0.30	0.49	0.30	6195
Weighted Avg	0.74	0.58	0.62	6195

Table 6.2: Classification Report - Rule-based System

system displays robust performance particularly in the time category, with an F1-score of 0.82 for *B-time*, and 0.88 for *I-time*. However, the performance on *B-participant* reveals a notable discrepancy between high precision (0.78) and low recall (0.09), resulting in an F1-score of 0.16. Compared to the GPT-prompting system, the fine-tuned BERT system excels greatly in recognising *I-participant*, Overall, the system maintains a weighted average F1-score of 0.87 and accuracy of 0.88, proving effective in detecting activity information across conversations.

	Precision	**Recall**	**F1-Score**	**Support**
B-event	0.67	0.67	0.67	193
B-participant	0.78	0.09	0.16	161
B-place	0.22	0.18	0.20	22
B-time	0.84	0.79	0.82	135
I-event	0.63	0.68	0.66	544
I-participant	0.64	0.78	0.70	9
I-place	0.56	0.52	0.54	42
I-time	0.86	0.90	0.88	288
O	0.92	0.94	0.93	4801
Accuracy			0.88	6195
Macro Avg	0.68	0.62	0.62	6195
Weighted Avg	0.87	0.88	0.87	6195

Table 6.3: Classification Report - Fine-tuned BERT System

6.1.3 GPT-prompting System

The GPT-prompting system which applies a zero-shot method powered by GPT-4o to label the dataset, stands out as the top performer among the three systems. It achieves a macro average precision of 0.68, recall of 0.73, and F1-score of 0.70. Notably, even the least effective category, *I-participant*, attains an F1-score of 0.50, which surpasses the corresponding F1-score in the other two systems. The system particularly excels in recognising temporal and event-related tokens (B-time, I-time, and I-event) with F1-scores of 0.78, 0.81, and 0.73 respectively.

	Precision	**Recall**	**F1-Score**	**Support**
B-event	0.68	0.69	0.68	193
B-participant	0.72	0.75	0.73	161
B-place	0.44	0.64	0.52	22
B-time	0.80	0.76	0.78	135
I-event	0.70	0.78	0.73	544
I-participant	0.57	0.44	0.50	9
I-place	0.46	0.74	0.56	42
I-time	0.81	0.82	0.81	288
O	0.95	0.93	0.94	4801
Accuracy			0.89	6195
Macro Avg	0.68	0.73	0.70	6195
Weighted Avg	0.90	0.89	0.89	6195

Table 6.4: Classification Report - GPT-Prompting System

Chapter 7

Error Analysis

In this chapter, the typical errors appearing in the three systems will be discussed. In all provided examples, the gold labels will be underlined, while the system-predicted labels will be indicated in **bold**.

7.1 Rule-based System

7.1.1 Overtagging

The classification report, as detailed in Table 6.2, highlights a discrepancy between the precision and recall metrics across various categories, with precision consistently lagging behind the recall. For instance, the recall for *B-event* and *B-participant* stands at 0.79 and 0.94 respectively, while their corresponding precision scores are only 0.16 and 0.31. Further examining the confusion matrix (Figure 7.1) reveals that a significant number of *O* tokens are mistakenly identified as event-related expressions. This tendency to misclassify non-event tokens as event-related demonstrates a prevalent issue of overtagging by the system.

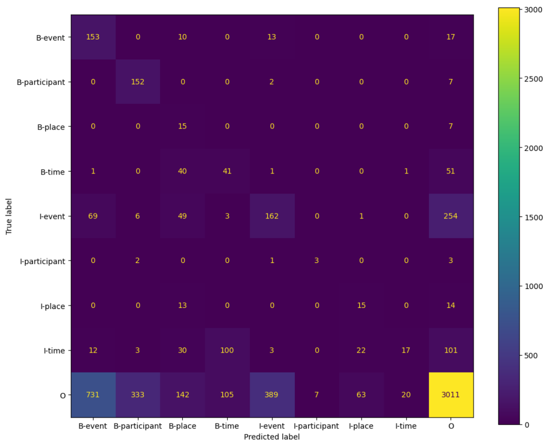

Figure 7.1: Confusion Matrix - Rule-based System

This overtagging is inherent to the nature of the rule-based system, where each sentence within a conversation is processed and tagged according to pre-defined rules. Consequently, when an event is mentioned across multiple turns in a conversation, it will be labelled repetitively.

7.1.2 Linguistic Rules Incompletion

After inspecting the model predictions and comparing them to gold labels, the errors caused by incomplete linguistic rules are discovered. For example, in this turn of conversation, the chatbot is attempting to engage by asking about the patient's garden, and the patient's response reveals a daily habit of taking care of the plants. However, within the rule-based framework, the nominal subject of the predicate is identified as the participant. Consequently, "your garden" is incorrectly labelled as the participant of the sentence, illustrating the limitation in the system's ability to correctly contextualise and interpret conversational data.

- *Chatbot: How is **your garden** doing these days?*
- *Patient: I tend to the plants every morning before breakfast.*

7.1.3 Lack of Context

Moreover, because the system follows strict linguistic rules, it lacks the capability to interpret contextual nuances, which can further lead to inaccurate tags. Such limitations highlight the challenges in rule-based systems, particularly in dynamic conversations where contextual understanding is crucial.

In the following instance, the rule-based system mislabelled "helping you" as the event, but this event is negated in the patient's response. It is because the system can only process the conversational data sentence by sentence, without grasping the context, which highlights its limitations in understanding and contextualising responses effectively.

- *Chatbot: Do you have anyone **helping you** with your plants?*
- *Patient: No, I prefer taking care of them myself.*

7.2 Fine-tuned BERT Model

7.2.1 Span Starts with I- Label

In the predictions of the fine-tuned BERT system, it has been observed that some spans incorrectly start with an I- (inside) label instead of a B- (begin) label. For instance, in the examples provided, the labels for "tidy" and "trying" should be *B-event*, yet the model outputs *I-event*.

- *I try to **tidy** up a bit each evening before I wind down for bed.*
- *Yes, I find joy in cooking new recipes and **trying** different flavors.*

This issue may arise because *B-event* (193 cases) labels in the training set are significantly less frequent than *I-event labels* (544 cases), causing the model to incorrectly default to using I- labels. Additionally, while BERT captures long-range dependencies, it may struggle with precisely identifying the boundaries of spans, contributing to this

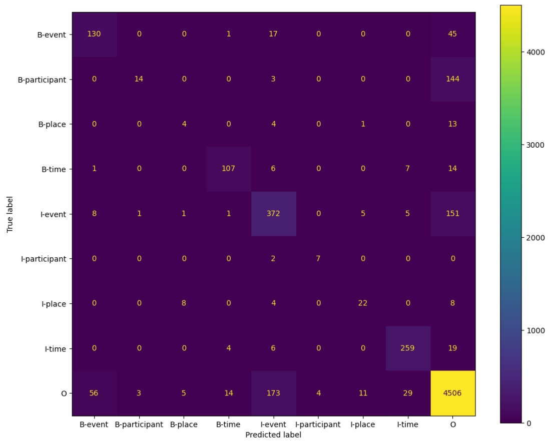

Figure 7.2: Confusion Matrix: Fine-tuned BERT Model

tagging error. It can be effective to mitigate this mislabelling issue by implementing post-processing procedures to correct such obvious errors.

7.2.2 Training Set Mislabelling

According to the confusion matrix (Figure 7.2), we can observe that 144 instances of *B-participant* are mislabelled as *O*, indicating that the system fails to recognise the participant token. This issue is further highlighted by the metrics for *B-participant*, with a precision of 0.78 and a recall of only 0.09 (Table 6.3). The problem arises because the BERT-based system was fine-tuned using GPT-4o labelled data, which omitted the labelling of *B-participant* in its preliminary version (discussed in Section 5.3).

- *I cook three times a day.*
- *I do my laundry every Saturday morning to keep everything organized.*
- *I prefer to do housework in the morning when I have more energy.*

A similar issue affects the prediction of the label *B-place*, which exhibits a recall of only 0.18. The model tends to omit the preposition indicating location, such as "in" in "in the backyard", because the training data does not include such prepositions in the locative phrase span. As a result, the *B-place* tag is frequently mislabeled.

- *I enjoy planting flowers and vegetables with my grandchildren in **the backyard**.*
- *How often do you cook meals at **home**?*
- *When do you usually take care of the plants in **your garden**?*

To address this issue, it is essential to employ the optimised version of the prompt described in Section 5.3 for generating more accurate labels with GPT-4o, which are then used in the fine-tuning process of the BERT model. This approach ensures the accuracy of the labels, thereby enhancing the model's overall performance.

7.3 GPT-prompting System

7.3.1 Temporal Expressions

Although the system performs well in labelling time expressions (F1=0.78 for *B-time*, F1=0.81 for *I-time*), it struggles with identifying time expressions in certain contexts.

Modifier Misinterpretation

In the following example, the system labels "for the week" as a time expression, but it actually functions as a modifier of the noun "vegetables", indicating that the pre-chopped vegetables are intended for use over the next week, rather than specifying the duration of the action "pre-chop" will last for a week.

- *Chatbot: How do yo make your food preparation easier for yourself?*
- *Patient: I usually pre-chop vegetables **for the week**.*

Similarly, the system incorrectly labels the phrase "lunch and dinner" as time expressions, whereas it actually serves as a modifier for "meals," specifying that the meals prepared are intended for lunch and dinner.

- *Chatbot: How was your day today?*
- *Patient: Oh, it was quite busy. I prepared meals for **lunch and dinner**.*

This error highlights the system's limitation in understanding the role of modifiers within a sentence. Consequently, the system can be enhanced by employing different prompting methods, such as few-shot prompting that specifically includes examples of modifiers.

Time-Place Confusion

The confusion between time and place expressions is observed between the system predictions and the gold labels. In the examples below, temporal phrases like "in the afternoons" and "on Sundays" are misclassified as locations. This error likely occurs because the system sometimes relies on surface pattern matching rather than deep semantic understanding, given that time and place expressions often share similar structures like "in the" and "on."

- *Chatbot: How has your routine been lately?*
- *Patient: I've been busy maintaining the garden **in the afternoons**.*

- *Chatbot: How do you usually plan your meals for the week?*
- *Patient: I plan my meals **on Sundays** and make a grocery list before going shopping.*

Time-Event Confusion

In addition, another type of error regarding time expressions has been identified. The system sometimes fails to extract time expressions from a longer span of events. In the first example, the system recognises "walk for 30 minutes every morning and do light stretches in the afternoon" as an event but does not identify "every morning" and "in the afternoon" as time expressions. Similarly, in the second example, the system recognises "unwind in the evening by taking off clothes" as an event but fails to identify "in the evening" as a time expression.

- *Chatbot: How often do you engage in physical activities?*
- *Patient: I walk for 30 minutes* ***every morning*** *and do light stretches* ***in the afternoon****.*

- *Chatbot: What about putting off clothes at the end of the day?*
- *Patient: I usually unwind* ***in the evening*** *by taking off clothes.*

7.3.2 Span Location Mismatch

Upon manual inspection, several cases that are mislabeled still hold semantic validity. In the following example, the system identifies "cook meals" in the chatbot's utterance as an event, whereas the gold label for the event appears in the patient's response. These instances are semantically correct but affect overall system performance.

- *Chatbot: How often do you* ***cook meals*** *at home?*
- *Patient: I <u>cook meals</u> almost every day, mainly for dinner.*

Similarly, in this example, the system labelled "cutting your nails" in the chatbot's utterance as the event, while the gold annotation for the event is in the patient's response: "trim my nails". The event is mentioned by both speakers, but the system's labelling does not align with the gold standard annotation, leading to a discrepancy.

- *Chatbot: Do you find time for* ***cutting your nails****?*
- *Patient: I <u>trim my nails</u> every two weeks to keep them neat and clean.*

To address this issue, the evaluation method should be enhanced to focus on the semantic meaning of the labels, rather than their exact location within the conversation.

Chapter 8

Discussion

8.1 Limitations

Although the model performance for the experimental groups is promising ($F1_{GPT} = 0.70$ and $F1_{BERT} = 0.62$), it has only been validated on synthetic conversational data. According to the A-PROOF team expert's evaluation, the synthetic data exhibits gaps when compared to real-life conversations, particularly in terms of naturalness, especially within the *domestic life* category. Therefore, the diversity of the conversations in the dataset might not adequately capture the range of real-world scenarios the model will encounter. When these models are applied to natural conversations, their effectiveness may not be as expected.

Also, the size of the dataset is limited due to the time constraints of this project. There are only 687 conversations in the training dataset and 54 in the development and test datasets. This limited amount of training data may not be adequate for the models to learn and generalise effectively, potentially hindering their performance. Furthermore, the small size of the development and test datasets restricts the comprehensive validation and evaluation of the models, thereby impacting the reliability and robustness of the performance assessments. Moreover, due to time constraints, the gold labels for the development and test data were annotated manually by the author. To improve the annotation reliability and consistency, it would be beneficial to invite multiple annotators to label the same datasets and calculate the Inter-Annotator Agreement (IAA) score.

In addition, in terms of the fine-tuned BERT model, the labels in the training dataset are generated by GPT-4o, which is proved to have disagreements with the manual annotation (Section 5.3). The inconsistencies in labelling led to sub-optimal model outcomes, as the model learned from misaligned data.

8.2 Future Work

8.2.1 Refining Dataset

For future work, it is crucial to expand the dataset to enhance the effectiveness of the multilingual BERT model. Currently, the dataset consists of only 5,965 utterances; increasing this number could significantly stabilise the fine-tuning process. Additionally, enlarging the development and test sets, which currently contain 468 and 478 utterances respectively, will enable more comprehensive evaluation and validation of the model's performance.

According to the findings from testing the GPT-prompting system with development data, the labels generated by GPT-4o often do not match the manual annotations. To address this issue, it may be beneficial to employ a few-shot method in prompt engineering or to refine the prompts to ensure they align with the same criteria as the manual annotations.

Additionally, the error analysis of the GPT-prompted system reveals that some mislabelled tokens share the same meaning as the gold annotations but differ in their placement within the conversation. It can be essential to improve the evaluation criteria, by assessing the extracted information based on its semantic meaning rather than its specific location in the conversation. Moreover, the system's predicted labels should undergo post-processing to eliminate common errors, such as those beginning with I-(inside) labels.

8.2.2 Structure Event-Argument Relationships

This project successfully extracts activity-related information within conversations. For future studies, it is essential to identify the relationship between the extracted events and their corresponding arguments (time, place, and participants). Similar to dependency-based semantic role labelling tasks, where the relationship between predicates and their arguments is determined after detecting and classifying all the predicates and arguments present ((Gildea and Jurafsky, 2002), (Carreras and Màrquez, 2005)), the relationships between events and event-related arguments can be realised in a structured manner. Specifically, it is feasible to enhance the understanding of how events are interconnected through their temporal, spatial, and participant contexts by integrating techniques from dependency-based semantic role labelling of PropBank ((Johansson and Nugues, 2008)).

8.2.3 Enrich Sentiment Information

Furthermore, upon inspecting the test dataset, it is evident that merely extracting the event, its time, place, and participants is insufficient for doctors to fully understand the patients' conditions, because patients often express positive or negative emotions while describing an event.

For example, when a patient says "I had a bit of a challenge when I climbed the stairs this morning", only extracting $I_{participant}$, $climbed\ the\ stairs_{event}$ and "$this\ morning_{time}$" is inadequate. The phrase "a bit of a challenge" conveys crucial information about the patient's emotional state and the difficulty they experienced. Therefore, future work should focus on incorporating sentiment analysis to enrich the extracted information, thereby providing doctors with a more detailed and accurate representation of the patient's conditions.

Chapter 9

Conclusion

This project explored the feasibility to extract activity-related information within conversations between patients and a healthcare chatbot. The conversations used in this project were generated by prompting GPT-4o, guided by the definitions for human activity provided by International Classification of Functioning, Disability, and Health (ICF). Following the discussions with healthcare experts in the A-PROOF team, all conversations were generated under the categories of *mobility*, *self-care*, and *domestic life*. Due to time constraints, the labels for the training set were generated by prompting GPT-3.5-turbo, while the development and test datasets were annotated manually by the author.

The experiment involved the development of three NLP systems: a rule-based system serving as the baseline, a fine-tuned BERT system, and a GPT-prompting system as the experimental groups. The rule-based system leveraged dependency and named entity recognition information to define rules for extracting event-related tokens. The BERT-based system fine-tuned a pre-trained multilingual BERT model using the GPT-generated training dataset. The GPT-prompting system employed zero-shot methods to generate activity information.

The results indicated that the GPT-prompting system performed the best. In addition, the fine-tuned BERT system also showed moderate performance, with its errors primarily stemming from inaccurate labels in the training set. The baseline rule-based system proved unsuitable for analysing dynamic conversational data. Error analysis also revealed that the evaluation criteria could be refined to assess outcomes based on semantic meaning rather than strict text spans.

Future studies should focus on developing a more comprehensive dataset for training, validation, and testing. Additionally, mapping event-argument relationships by employing dependency-based methods, which have been validated in semantic role labeling tasks is also essential. Furthermore, incorporating more detailed information, such as the sentiment of each event, would provide a more comprehensive understanding of the patients' conditions for the doctors.

Bibliography

S.-K. Agnieszka, I. MARCINKOWSKA, A. RUZIK-SIERDZIŃSKA, and R. MACKEVIČIŪTĖ. Ageing policies-access to services in different member states. 2021.

E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell. On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623, 2021.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

K. Bu, Y. Liu, and X. Ju. Efficient utilization of pre-trained models: A review of sentiment analysis via prompt learning. *Knowledge-Based Systems*, page 111148, 2023.

X. Carreras and L. Màrquez. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of the ninth conference on computational natural language learning (CoNLL-2005)*, pages 152–164, 2005.

K. V. Deemter, M. Theune, and E. Krahmer. Real versus template-based natural language generation: A false opposition? *Computational linguistics*, 31(1):15–24, 2005.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

A. Gatt and E. Krahmer. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170, 2018.

D. Gildea and D. Jurafsky. Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288, 2002.

J. Guralnik, E. Simonsick, L. Ferrucci, R. Glynn, L. Berkman, D. Blazer, P. Scherr, and R. Wallace. A short physical performance battery assessing lower extremity function: Association with self-reported disability and prediction of mortality and nursing home admission. *Journal of gerontology*, 49:M85–94, 04 1994. doi: 10.1093/geronj/49.2.M85.

M. Honnibal and I. Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7 (1):411–420, 2017.

F. Huang, A. Ahuja, D. Downey, Y. Yang, Y. Guo, and A. Yates. Learning representations for weakly supervised natural language processing tasks. *Computational Linguistics*, 40(1):85–120, 2014.

T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

R. Johansson and P. Nugues. Dependency-based semantic role labeling of propbank. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 69–78, 2008.

Y. Liu and M. Lapata. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*, 2019.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

A. McCallum, K. Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Madison, WI, 1998.

J. Nivre, M.-C. De Marneffe, F. Ginter, Y. Goldberg, J. Hajic, C. D. Manning, R. McDonald, S. Petrov, S. Pyysalo, N. Silveira, et al. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, 2016.

W. H. Organization. International classification of functioning, disability and health : Icf, 2001.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

E. Reiter and R. Dale. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87, 1997.

K. Salter, J. Jutai, R. Teasell, N. Foley, J. Bitensky, and M. Bayley. Issues for selection of outcome measures in stroke rehabilitation: Icf participation. *Disability and rehabilitation*, 27(9):507–528, 2005.

F. Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units, 2016.

M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism, 2020.

L. N. Smith. No more pesky learning rate guessing games. *CoRR*, abs/1506.01186, 2015. URL http://arxiv.org/abs/1506.01186.

Statistics Netherlands (CBS). Elderly people - cbs. https://www.cbs.nl/en-gb/visualisations/dashboard-population/age/elderly-people, 2024.

E. Strubell, A. Ganesh, and A. McCallum. Energy and policy considerations for modern deep learning research. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13693–13696, 2020.

M. Surdeanu, R. Johansson, A. Meyers, L. Màrquez, and J. Nivre. The conll 2008 shared task on joint parsing of syntactic and semantic dependencies. In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 159–177, 2008.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023.

A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht. The marginal value of adaptive gradient methods in machine learning. *Advances in neural information processing systems*, 30, 2017.

J. L. Wolff, B. Starfield, and G. Anderson. Prevalence, expenditures, and complications of multiple chronic conditions in the elderly. *Archives of internal medicine*, 162(20): 2269–2276, 2002.