

Master Thesis

Usage of Generative Models to Ask Follow-up Questions for Health Monitoring

Yijing Zhang

*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 Th.J.M. Vossen
2nd reader: Sophie Arnoult

Submitted: August 27, 2024

Abstract

This project explores the development of an automated method for generating effective follow-up questions aimed at monitoring patients' physical and mental conditions using the Llama3 generative model. Two primary optimization strategies were evaluated: prompt engineering (employing zero-shot and few-shot prompting) and model fine-tuning. The study contrasted fine-tuned models, without prompt engineering, with non-fine-tuned models that applied prompt engineering techniques. The primary finding is that fine-tuned LLaMA3 models significantly outperform non-fine-tuned models, demonstrating its superior effectiveness of task-specific adaptation over prompting methods. This highlights the advantage of fine-tuning in generating more accurate and relevant follow-up questions for patient monitoring.

Declaration of Authorship

I, Yijing Zhang, declare that this thesis, titled *Usage of Generative Models to Ask Follow-up Questions for Health Monitoring* 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: 08/27/2024

Signed:

Acknowledgments

I would like to express my sincere gratitude to Piek Vossen for his supervision and instant feedback, and to my second reader, Sophie Arnoult, for her critical analysis of this work. I am deeply thankful to the A-proof Team and the healthcare experts: Edwin, Marike and Sabina for their invaluable assistance and patience in creating the test data. I also would like to thank my friend, Danna, who has accompanied me during the thesis writing with her excellent cooking skills. The last but not the least, I would like to thank my parents. They have expressed their complete understanding of my situation, ready to try their best to support me no matter what happens.

List of Figures

2.1	examples.ICF layers	5
3.1	Number of Activity Types for Each Category	15
3.2	Distribution of Activities across Categories	18
3.3	Activity Distribution in Communication	18
3.4	Activity Distribution in Self-care	18
3.5	Activity Distribution in Mobility	19
3.6	Conversation Distribution of Test	23
3.7	FQs Distribution of Test	23
4.1	System Comparison of F1 Scores	36
4.2	F1 Scores of Cate.FQ type across Systems	38
4.3	F1-Scores for Each Model and Average F1-Scores for Each Category	39
4.4	Comparison of Average BERT F1 Scores for Emotion and Function FQ Types across Different Systems	40

Contents

Abstract	i
Declaration of Authorship	iii
Acknowledgments	v
List of Figures	vii
1 Introduction	1
2 Related Work	3
2.1 ADL and Ontology	3
2.2 Question Generation and LLMs	6
2.3 Prompt Learning and Engineering	7
2.3.1 Zero-shot and Few-shot prompting	7
3 Methodology	9
3.1 Data Collection	9
3.1.1 Data Ontology	9
3.1.2 Prompt Engineering	10
3.1.3 Zero-shot and Few-shot prompting	13
3.1.4 Experimental Set-up	14
3.1.5 Prompt Evaluation	16
3.2 Data Statics and Preprocess	16
3.2.1 Data Preprocess and Statics	16
3.2.2 Training Data Statics and Format	17
3.2.3 Validation and Test Data	22
3.3 Experimental Set-up	25
3.3.1 Model Introduction: LLaMA3	25
3.3.2 Model Parameter Tuning	25
3.3.3 Optimization Approaches	26
3.4 Evaluation	27
3.4.1 Data Post-process	27
3.4.2 Automatic evaluation	30
3.4.3 Human evaluation	31
4 Results and Analysis	33
4.1 Results	33
4.2 System-wise Comparison	36
4.3 Category-wise Comparison	37
4.4 FQ type-wise Comparison	39

5	Error Analysis and Discussion	43
5.1	Error Analysis	43
5.1.1	Errors in Identifying Target Activities	43
5.1.2	Errors in Generating Task-Objective-Relevant FQs	46
5.1.3	Relevant, Complete and Coherent	47
5.2	Discussion of Results	48
5.2.1	Limitation	48
5.2.2	System-wise	48
5.2.3	Category-wise	49
5.2.4	FQ type-wise	50
6	Conclusion and Future Work	51

Chapter 1

Introduction

Monitoring people’s health and well-being before and after the treatment is important, which helps the prevention of severe illness. Such monitoring necessitates the continual collection of long-term data on individuals’ daily activities and emotions, extending beyond direct medical care. However, the geographical and temporal barriers may stop people from getting monitoring and receiving treatment in time.

One approach to overcoming these challenges is the development of automated monitoring systems using conversational agents (CAs), such as chatbots, to conduct daily online conversations with users, thereby bypassing geographical and temporal obstacles and to collect information, where CAs can ask questions, interpret responses and probe answers (Xiao et al., 2020b). CAs have demonstrated their ability to gather information and perform information elicitation tasks in various fields, including public service Makasi et al. (2020); Stamatis et al. (2020); Shum et al. (2018), education Hiremath et al. (2018); Heilman and Smith (2010); Su et al. (2018); Clarizia et al. (2018), and healthcare Athota et al. (2020); Xu et al. (2021); Lee et al. (2020).

In the healthcare domain, Vossen et al. (2024) proposed a project utilizing a Hybrid Intelligence Agent to create a personal diary through conversations with patients, contextualizing their physical, social, and mental functioning and well-being for medical research and monitoring. This monitoring primarily involves detecting Activities of Daily Life (ADL) within conversations, with a communicative agent regularly engaging stakeholders to gather more detailed information through follow-up questions (FQs). They argue that such a CA-based approach can effectively provide context for monitoring both physical functioning and mental well-being.

Inspired by this project, a similar process can be employed to monitor individuals’ health. This process can be divided into two steps: First, caretakers initiate conversations with patients about their daily activities, focusing on attributes such as timing, location, and participants. Second, caretakers ask follow-up questions related to these activities to elicit responses that provide functional information and insights into the patients’ emotions, which can be categorized as positive or negative sentiment. In other word, there are two types of follow-up questions. The first is FQs on function-level information. This type of question can ask for information on more specific questions related to the target activity and evoke answers reflecting the function level. The second is FQs on emotional feedback. This type of question is asked from the perspective of the users about their positive or negative feeling of the activity.

For example, consider a conversation about the activity of ”dressing” between a caretaker (C) and a patient (P), generated by medical experts to closely resemble natural and real interactions:

C: Good morning! How did dressing go for you this morning?

P: Oh, it took a bit longer than usual today, but I managed to get my favorite outfit on.

To monitor the function level, the following follow-up questions might be asked:

C: Oh, so you managed it without help?

P: Yes, I did.

C: Why did it take more time than usual?

P: Because I wanted to wear a blouse with buttons.

C: What's tricky about that?

P: I have trouble getting the buttons to close.

For assessing emotional perspectives, the follow-up questions might include:

C: How do you feel about dressing for a bit longer than usual today?

P: Well, I have trouble wearing a blouse with buttons, but it's part of age. Fine motor skills are not what they used to be.

C: So it doesn't really bother you?

P: No, I just take more time for it and don't wear a blouse with buttons every day.

By asking whether the patient managed dressing without help and why it took longer than usual, the caretaker may detect changes in the patient's self-care abilities for function level monitoring. Additionally, by inquiring about the patient's emotional response to the activity, the caretaker can monitor the patient's mental state.

This approach allows caretakers to track potential changes in physical and mental conditions through rich, regular, and continuous conversational data, which can trigger alerts for further attention. Automating this process also has the potential to reduce the burden on the healthcare system.

To construct such a system, it is crucial to obtain the necessary information for monitoring. Therefore, CAs must ask the "right" follow-up questions about daily activities, probing and prompting more information based on conversation history and people's responses. Specifically, the FQs should be relevant to ask functional levels and mental states, such as the difficulty of performing activities and emotional responses to them. Although researchers have developed various techniques to enhance CAs' information elicitation abilities Lee et al. (2020); Motalebi et al. (2019), CAs still struggle to identify important concepts or events in conversations that could be explored further, and they lack the skills to ask pertinent follow-up questions Hu et al. (2024); Xiao et al. (2020a).

This project aims to explore an automatic method to generate "right" follow-up questions for monitoring patients' physical and mental conditions by employing the generative model of Llama3, with two optimization methods: prompt engineering with zero-shot and few-shot prompting and fine-tuning the model. That is, there are primarily two systems in the project to be compared: the fine-tuned (**FT**) systems without prompt engineering and the non fine-tuned (**non-FT**) systems with prompt engineering.

The thesis is structured as follows: Chapter 2 introduces related work on ontology, question generation, and prompt engineering. Chapter 3 of Methodology details the research design, including data preparation methods, experimental setup, and evaluation. Chapter 4 presents the experimental results and provides an analysis of these results. Finally, Chapter 5 covers the discussion with error analysis and Chapter 6 concludes the whole project.

Chapter 2

Related Work

This section introduces the related work of background and methodology used in the project. The first part is about ontology which can be used to determine the scope of conversations and FQs interpretation. The second part gives a related work of natural language generation (NLG), specifically question generation (QG). The third part provides related work of prompt engineering.

2.1 ADL and Ontology

Ontology is needed in the current project to represent a structured framework of daily activities with their definitions and give the reference to the corresponding function.

According to the Stanford Encyclopedia of Philosophy, activities are defined as intentionally performed events aimed at achieving specific purposes. These activities can be categorized into various types, such as educational or communicative, depending on their objectives. This task focuses on Activities of Daily Living (ADLs), which include the essential skills required for independent living and serve as indicators of an individual's functional status, such as eating, bathing, and mobility (Edemekong et al., 2019). The inability to perform these essential ADLs can lead to unsafe conditions and a reduced quality of life. Therefore, assessing an individual's ability to manage ADLs is crucial, as it can indicate the need for assistance and further evaluation (Edemekong et al., 2019). ADLs are generally classified into basic (BADLs) and instrumental categories, with the current task emphasizing BADLs due to their fundamental importance.

- **Basic Activities of Daily Living (BADLs):** These are fundamental self-care tasks necessary for maintaining independence and physical well-being, For examples:
 - Bathing: Washing oneself, including the whole body or specific parts.
 - Dressing: Selecting appropriate clothing and putting it on.
 - Eating: The ability to feed oneself.
 - Mobility: Moving around, including walking and transferring (e.g. from a bed to a chair).
 - Toileting: Managing personal hygiene and using the toilet.
 - Continence: Controlling bladder and bowel functions.
- **Instrumental Activities of Daily Living (IADLs):** These are more complex activities that are necessary for living independently but are not essential for basic functioning. For examples:
 - Managing Finances: Handling money, paying bills.

- Medication Management: Taking medications as prescribed.
- Cooking: Preparing meals.
- Shopping: Purchasing groceries or other necessities.
- Housekeeping: Cleaning and maintaining the home.
- Using Transportation: Driving or using public transportation.

To effectively measure functional status through activities, a common and universally accepted framework and classification system is necessary. The WHO International Classification of Functioning, Disability, and Health (ICF) provides such a framework. It is designed to measure health and disability at both individual and population levels, offering a standardized language and reference model for describing health and related states (Cozzi et al., 2021).

According to Stucki et al. (2008), the International Classification of Functioning, Disability, and Health (ICF) defines "mutually exclusive and cumulatively exhaustive" categories, which can serve as a reference for various medical purposes, including the development of clinical and self-reported measurement instruments. The ICF is closely linked to functional levels that reflect health conditions, which can be quantified using the Linkage Methodology.

The Linkage Methodology is a systematic approach used to map health-related information—such as data from clinical assessments, patient-reported outcomes, or qualitative studies—to the ICF framework. This methodology plays a crucial role in standardizing and interpreting diverse health-related data within the universal framework provided by the ICF. The methodology involves two main steps: first, identifying meaningful concepts within the health-related information that needs to be translated to the ICF; second, linking these identified concepts to the most appropriate ICF categories according to established linking rules. These rules ensure that concepts are accurately mapped to ICF categories that best represent them, whether they pertain to body functions, activities and participation, environmental factors, or personal factors.

The Linkage Methodology facilitates the integration of various health-related data into a standardized language, allowing for the comparison of functioning across different measures, conditions, and contexts. It also supports the development of ICF-based measurement instruments, such as the ICF Checklist and ICF Core Sets. Additionally, it enables the transformation of information from clinical tests or patient-oriented instruments into the ICF qualifier scale, which is a rating scale for assessing the extent of problems in functioning.

Based on this methodology, the ICF serves as a reference framework for quantifying functional levels (Kelvin, 2008). The ICF is divided into two main components: ICF Categories and ICF Qualifiers.

ICF Category has four main categories: Body function, Activities and participation, Environmental factors, and Body structure. Daily activities (BADLs) concerning the current task mainly are included in Activities and Participation. Therefore, only Activities and Participation will be considered in the current task (referred as "main category").

Within Activities and participation, there are 9 categories of *Learning and applying knowledge, Learning and applying knowledge, General Task and demands, communication, Mobility, Self-care, Domestic Life, Interpersonal interactions and relationships, Major life areas, Community, social and civic life* (Organization, 2001). Under each category, there are two to three layers to show the sub-categories, examples shown as Figure 2.1. To be more specific, taking one of the category series for example, indicted by red cells in Figure 2.1: category: *Mobility* - 1st layer: *Walking and Moving* - 2nd layer: *Moving around*- 3rd layer: *Jumping*.

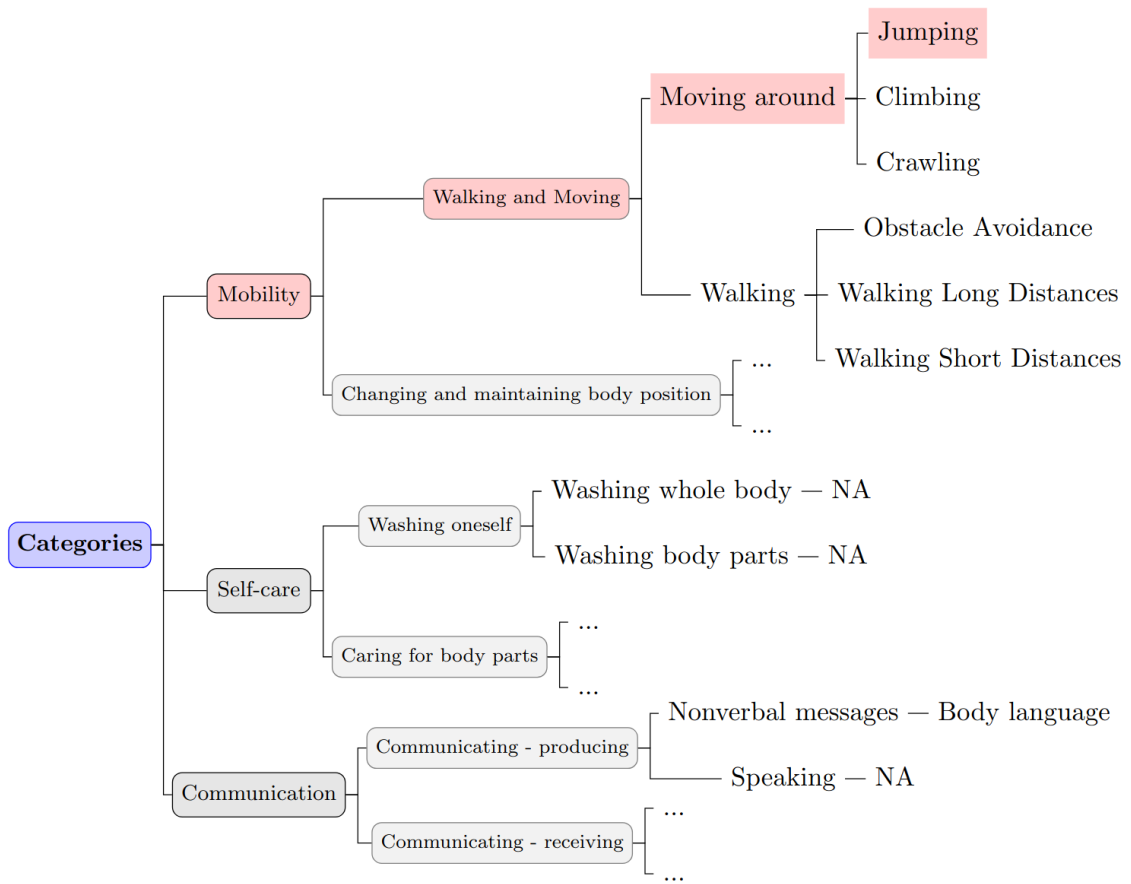


Figure 2.1: examples_ICF layers

ICF Qualifier is a generic scale to quantify a single ICF category in terms of five aspects: *Barrier or facilitator, Capacity, Extent or magnitude of impairment, Performance, Nature of change in body structure*. Each aspect can be quantified by mainly 5 levels, shown as the Table 2.1,

ICF Qualifier	Percentage of Problem
0-NO difficulty (none, absent, negligible)	0-4%
1-MILD difficulty (slight, low)	5-24%
2-MODERATE difficulty (medium, fair)	25-49%
3-SEVERE difficulty (high, extreme)	50-95%
4-COMPLETE difficulty (total)	96-100%

Table 2.1: ICF Qualifiers and their Corresponding Percentage of Problem

There are two methods for measuring functional levels using the ICF framework. The first method employs a rating scale ranging from 0 to 4 as a quantifier. The second method involves transforming clinical or medical information into these quantifiers, thereby allowing the function level to be quantified on the rating scale.

This ontology links activities or events in conversations to functional levels and will be utilized for generating conversation data and functional questionnaires (FQs) when preparing the training data (see Chapter 3: Methodology).

2.2 Question Generation and LLMs

Research on questions has been a significant focus in Natural Language Generation, primarily aiming at answering questions as an intermediate step in the question-answering process. However, Question Generation (QG) refers to the process of automatically generating questions from various types of input, including databases, deep semantic representations, and raw text (Zhang et al., 2021; Rus et al., 2011, 2010). Unlike question-answering, QG is evaluated based on the intrinsic quality of the generated questions (Heilman and Smith, 2010), rather than the answers that follow.

There are two main approaches in the field of Question Generation (QG): rule-based methods and neural network-based methods (Hu et al., 2024).

Rule-based methods rely on handcrafted rules to transform given texts into questions and can be broadly classified into template-based, syntax-based, and semantics-based approaches (Zhang et al., 2021). These methods are straightforward and offer greater control over the output with minimal data requirements. However, they depend heavily on manually created rules, which can limit the variety of generated questions and reduce adaptability to different topics or domains.

In contrast, neural network-based methods have recently advanced significantly, driven by data-driven algorithms and access to large-scale datasets. These methods provide an end-to-end trainable QG framework, enabling joint optimization of content selection and question generation, which is a common challenge in QG tasks. Compared to rule-based methods, neural network-based approaches excel in generating questions with greater fluency, flexibility, and diversity (Hu et al., 2024; Zhang et al., 2021). Among these, large-scale pre-trained language models (LLMs) like GPT and LLaMA3 have achieved state-of-the-art results across various natural language generation (NLG) tasks, including question generation (Devlin et al., 2019; Zhang et al., 2021; Howard and Ruder, 2018).

LLMs are trained on vast and diverse datasets using self-supervised learning techniques. They encode input text into high-dimensional vector spaces, preserving semantic relationships between words and contexts, which allows them to learn statistical patterns and generate coherent, contextually relevant questions (Chen et al., 2024).

The quality of text generated by large language models (LLMs) is influenced by several factors, including the prompt provided, the model’s hyperparameters, and the diversity of the training data (Chen et al., 2024). For the current task, the primary focus is on the prompt used to guide the model, as LLMs are typically trained with well-tuned hyperparameters that have been optimized through extensive research and experimentation (Goodfellow et al., 2016). Additionally, models like GPT are trained on large, diverse datasets compiled from various sources across the internet, making them particularly well-suited for generating general conversations rather than domain-specific content.(Radford et al., 2019).

2.3 Prompt Learning and Engineering

Fine-tuning pretrained models is often costly and time-consuming, requiring substantial amounts of annotated data and computing resources. To mitigate these challenges, researchers have increasingly relied on prompts to guide model learning (Liu et al., 2021). A prompt serves as the model’s input, and variations in its syntax (e.g., the ordering of examples) and semantics (e.g., example selection, instructions) can significantly influence the model’s performance.(Arora et al., 2022; Chen et al., 2024; Kaddour et al., 2023; Lu et al., 2022).

Prompt learning is an emerging paradigm in natural language processing (NLP) that enables models to perform few-shot or even zero-shot learning, allowing them to adapt to new scenarios with minimal labeled data (Liu et al., 2021). This approach leverages language modeling techniques to directly model the probability of text sequences. The key to successful prompt engineering lies in designing prompts that effectively guide pretrained models to perform specific tasks.

Modifying the syntax and semantics of a prompt is crucial for steering LLM responses in the desired direction. This process, known as ”prompt engineering”, involves designing natural language queries that guide the model’s outputs toward specific outcomes (Kaddour et al., 2023). Current popular prompting methods range from foundational approaches like role-prompting (Shanahan et al., 2023) to more advanced techniques such as ”chain-of-thought” prompting (Wei et al., 2023).

Building on insights from Liu et al. (2021) and Wang et al. (2024), the following section outlines the three key steps in prompt learning:

1. Pretrained Models: Select an appropriate pre-trained model.
2. Prompt Engineering: Design prompts tailored to the specific requirements of each downstream task.
3. Task-Specific Response Design: Allow models to produce the desired output based on the task at hand.

These steps provide a guideline for creating prompts used in generating training datasets. The following section outlines two important methods used in prompt engineering: zero-shot prompting and few-shot-prompting.

2.3.1 Zero-shot and Few-shot prompting

Zero-shot prompting is a technique in natural language processing (NLP) where a model, such as a large language model (LLM), performs a task without any specific examples or prior training related to that task. In zero-shot prompting, the model is given only an instruction or prompt describing the task and generates responses based solely on its extensive pre-trained knowledge. LLMs like GPT or LLaMA, which are trained on vast and diverse datasets, are

capable of generalizing across various tasks and domains. This approach can yield impressive results when the prompt is well-designed, leveraging the model’s broad understanding and capability to infer task requirements (Wang et al., 2024).

However, zero-shot prompting has its limitations. The absence of specific examples can result in less accurate outputs, as the model relies heavily on its general knowledge and may struggle with ambiguities or specific nuances of the task. Without examples, the model’s responses might lack precision or relevance, especially for complex or domain-specific tasks.

To overcome these limitations, few-shot prompting has been introduced as a complementary approach. Few-shot prompting involves providing the model with a small number of example input-output pairs within the prompt. This additional context helps guide the model’s responses, enhancing its performance by illustrating the desired task behavior (Zhao et al., 2021).

In few-shot prompting, the model receives a prompt that includes a task description and several examples demonstrating how to handle specific inputs. For instance, if the task is text classification, the prompt may include a few labeled examples showing different types of inputs and their corresponding categories. These examples help the model better understand the task and generate more accurate responses for new, unseen inputs by building on the patterns learned from the provided examples. One of the primary benefits of few-shot prompting is its efficiency. By using only a few examples, this technique enables rapid adaptation to new tasks without extensive retraining or fine-tuning. The model leverages its pre-existing knowledge, acquired during its initial training, to perform well with minimal additional data. Few-shot prompting also helps reduce computational costs. Since it does not require updating the model’s parameters or conducting full-scale retraining, it minimizes the computational resources and time typically associated with fine-tuning. This makes few-shot prompting particularly suitable for scenarios where data is limited or quick deployment is essential.

Chapter 3

Methodology

This research mainly compares zero-shot prompting, few-shot prompting and fine-tuning approaches in generating relevant follow-up questions for monitoring the function level of users and their mental conditions via their daily activities.

The primary system comparison is between the fine-tuned (**FT**) systems and non fine-tuned (**non-FT**) systems. Fine-tuned systems include fine-tuning llama3 on datasets generated by GPT-3.5 using zero-shot prompting and few-shot prompting. After fine-tuning, llama models were tested using simple instructions, without employing additional few-shot or zero-shot prompts during the inference. On the other hand, the non fine-tuned (**non-FT**) models were tested using few-shot or zero-shot prompts during the inference.

This chapter first introduces the Data section, which covers Data Collection, including the use of data ontology and prompt engineering, as well as Data Statistics and Preparation. Next, it details the Experimental Set-up, which includes Data Statistics, Model Training, and Methodological Approaches. Finally, the chapter addresses Evaluation.

3.1 Data Collection

Fine-tuning large language models (LLMs) requires a substantial amount of data, including (1) conversations between caregivers and patients about an activity, and (2) two types of follow-up questions based on these conversations. Due to privacy concerns for medical data, it is impractical to use real conversations between caregivers and patients. Additionally, the extensive data requirements combined with time constraints make manual dataset creation challenging.

To address this issue, recent research has explored the potential of LLMs to generate synthetic data tailored for specific tasks, which has proven effective in various scenarios (Ye et al., 2022). Consequently, the current approach employs LLMs to generate training data, guided by carefully designed, task-specific prompts.

The Data section focuses on two key aspects of this process: prompt design and the selection of LLMs for data generation.

3.1.1 Data Ontology

The training data consists of two components: conversations and follow-up questions. Conversations are used for training, validation, and testing, while follow-up questions are used exclusively during the training phase. Both types of data will be generated based on ICF categories.

The International Classification of Functioning, Disability, and Health (ICF) is used in this task as an ontology for defining activities related to function levels and for quantifying functional levels (Stucki et al., 2008). As detailed in Chapter 2: Related Work, the ICF

framework is divided into ICF Categories and ICF Qualifiers. The process of generating conversations and follow-up questions is inspired by the Linkage Methodology (Stucki et al., 2008), which involves creating a list of ICF categories that mirrors the content of the original health-related information (such as clinical conversations). According to (Stucki et al., 2008), Linkage Methodology involves two main steps: first, identifying concepts within the health-related information for translation into ICF categories; and second, linking these concepts to the ICF.

In contrast, my approach reverses this process. Rather than starting with health-related concepts and mapping them to ICF categories, I begin with predefined ICF categories and their definitions. Using these categories as a guide, the model is tasked with generating associated concepts or events and creating conversations around these concepts.

The following is an example of linkage procedure 3.1,

Partial Medical Conversations	Identified Concepts	ICF Categories
My nails break more. I used to have long, strong nails, but now they break easily.	Breaking nails	Caring for finger-nails

Table 3.1: Linkage Procedure

I use the method reversely to generate the conversations. This reverse approach involves the following steps:

1. **Provide ICF Categories and Definitions:** I supply the model with specific ICF categories along with their definitions. This provides a clear framework and context for the types of concepts and events that should be generated.
2. **Generate Related Concepts and Events:** The model then produces concepts or events that are relevant to the given ICF categories. This step ensures that the generated content is aligned with the categories and reflects their scope and intent.
3. **Create Conversations:** The model generates conversations centered around the concepts or events derived from the ICF categories. These conversations are designed to be realistic and contextually relevant, simulating real-world interactions.

This process can also be illustrated by Table 3.2

ICF Categories	Generated Concepts	Generated Conversations
Caring for finger-nails	Breaking nails	My nails break more. I used to have long, strong nails, but now they break easily.

Table 3.2: Conversation generation process

By applying the Linkage Methodology in reverse, the generated conversations are health-related and align with ICF categories in content. This approach ensures that the conversations are relevant and can be used effectively for quantifying functional levels.

3.1.2 Prompt Engineering

Due to time constraints, the task focuses on three ICF categories: "communication", "mobility", and "self-care".

For LLM generation, the second layer of ICF categories is used. This layer is preferred over the first layer because it provides more specific details for generating conversations with

targeted events. Although the third layer offers even greater specificity, it can be overly detailed for daily conversation generation purposes and not all categories, such as "self-care", have a third layer. Thus, the second layer is chosen for its balance of specificity and manageability. Descriptions corresponding to these second-layer categories can be used as activity descriptions and examples to supplement the generation process.

The prompt content used for conversation generation is thus written as,

"Generate one small and natural online conversation about one activity. The conversation has around 4 to 6 utterances in total and each utterance should be completed and has less than 20 tokens.", "The care taker asks about the {activity}, which means {activity_description},{example_str}"

{activity}, {activity_description} and {example_str} are placeholders. The first will be filled in with second layer of ICF categories, {activity_description} will be filled in with the definition, {example_str} will be filled in with joined third layer. Table 3.3 provide example fillers for prompt content.

Category (Function)	1st Layer	2nd Layer (Activity)	Description
Mobility	Walking and Moving	Walking	Definition: Moving along a surface on foot, step by step, so that one foot is always on the ground. Examples: Strolling, sauntering, walking forwards, backwards, or sideways.

Table 3.3: Conversation prompt content

The prompt is then generated as,

"Generate one small and natural online conversation about one activity. The conversation has around 4 to 6 utterances in total and each utterance should be completed and has less than 20 tokens.", "The care taker asks about the 'Walking', which means 'Moving along a surface on foot, step by step, so that one foot is always on the ground', The activities can be about ¹ 'Strolling, sauntering, walking forwards, backwards, or sideways' "

After generating the conversations, these will serve as conversation history for LLMs to generate follow-up questions (FQs).

ICF functioning categories can be quantified using five levels. To better align the data with training models to ask function level-related questions, the "No difficulty" situation has been excluded. Consequently, the rating scale is revised into four levels: Mild difficulty (slight, low), Moderate difficulty (medium, fair), Severe difficulty (high, extreme), and Complete difficulty. Additionally, since follow-up questions typically seek more detailed information about activities to gain insights into function levels, the third layer of ICF categories, along with their descriptions, is included as supplementary details. Therefore, the prompts for generating FQs related to function levels are formatted as follows and {sub_activity} will be filled in with third layer of ICF categories along with the description. :

"Follow-up questions can ask more details about {sub_activity}, which must be able to evoke answers informing about the function level, such as questions evoking answers about mild, moderate, severe, or complete performance difficulty."

2nd Layer (Activity)	3rd layer (Sub-activity)	Description
Walking	Walking short distances	Walking for less than a kilometre, such as walking around rooms or hallways, within a building or for short distances outside.
	Walking on different surfaces	Walking on sloping, uneven, or moving surfaces, such as on grass, gravel or ice and snow, or walking aboard a ship, train or other vehicle.

Table 3.4: FQs prompt content

Table 3.4 provide examples to fill in the prompt. And the resulting prompt is,

”Follow-up questions can ask more in details about ‘Walking for less than a kilometre, such as walking around rooms or hallways, within a building or for short distances outside’, which must be able to evoke answers informing about the function level, such as questions evoking answers about mild, moderate, severe, or complete performance difficulty.”

Another type of FQs on emotional feedback is not based on the ontology. It can be texted as the example below,

”Follow-up questions evoke answers informing about emotional feedback, such as questions evoking answers about positive or negative feelings about the activity”.

Role-prompting is a fundamental method in prompt engineering(OpenAI), where the model is given a role to comprehend the input and generate the output (Zhang et al., 2023; Chen et al., 2024), such as a care taker or a patient. This method can be particularly effective in guiding the model’s responses and ensuring that they align with the desired output. For instance, if the prompt requires the model to play the role as a medical expert, LLMs have the potential to act more like an expected agent, providing a more precise and detailed response when asked about a medical condition or treatment (Xu et al., 2023).

In the current task, the model will be instructed to play the roles of care-takers, starting conversations and asking questions, and the roles of patients responding. The instruction of playing roles will also be given to the LLMs. Taking the prompt of generation conversation as an example,

```
query = [
{"role": "system", "content": "You need to play the roles of a care taker (C) and an elderly patient (P)"},
{"role": "system", "content": "Generate one small and natural online conversation about one activity. The conversation has around 4 to 6 utterances in total and each utterance should be completed and has less than 20 tokens."},
{"role": "system", "content": f"""The care taker asks about the activity of {activity}, which means {definition},{example}.
```

The format is as below,

C: starting conversation (start the conversation around the activity)

¹”The activities can be about” is used for coherent prompt when examples of this activity exists

P: utterance (respond naturally)

C: starting conversation (start the conversation around the activity)

P: utterance (respond naturally) ..."}],

```
{ "role": "user", "content": f"The topic of the conversation is about {activity}
and the patients can describe their daily activities and answer questions natu-
rally." }
```

Cloze prompts and **prefix prompts** are two types of prompt formats (Wang et al., 2024). Cloze prompts contain placeholders to be filled in the middle of the template, whereas prefix prompts refer to the input text being entirely situated before the generated output. For example, in the current prompt, part of can be formatted as,

```
query = [ { "role": "system", "content": f"" When the given conversation history
is: "{conversation}", follow-up questions {sub_activity} must be able to evoke
answers informing about the function level, such as questions evoking answers
about mild, moderate, severe, or complete performance difficulty."" },
```

```
{ "role": "user", "content": f"The patients can answer naturally and describe the
performance difficulty of performing {activity}. The performance difficulty can
be slight, fair, severe, or complete." },
```

```
{ "role": "system", "content": f"" the following up questions on monitoring func-
tion level can be"" }
```

or

```
query = [ { "role": "system", "content": f"" Follow-up questions {sub_activity}
must be able to evoke answers informing about the function level, such as ques-
tions evoking answers about mild, moderate, severe, or complete performance
difficulty."" },
```

```
{ "role": "user", "content": f"The patients can answer naturally and describe the
performance difficulty of performing {activity}. The performance difficulty can
be slight, fair, severe, or complete." },
```

```
{ "role": "system", "content": f"" When the given conversation history is: "{conversation}",
the following up questions on monitoring function level can be"" }
```

In order to make the input conversation more stand out and thus make the output FQs more relevant to the input, the current approach opts for the prefix format where the input is followed by the output immediately.

3.1.3 Zero-shot and Few-shot prompting

As mentioned in Related Work, Zero-shot prompting is tasking the model with something without any examples. Few-shot prompting is providing a few examples to guide the model in completing a task.

In the current task, both zero-shot and few-shot prompting methods are utilized. While traditional zero-shot prompting typically involves providing the model with only an instruction or prompt that describes the task, our approach slightly deviates from this standard definition. In our implementation of zero-shot prompting, we not only include the basic task instructions but also integrate background knowledge related to ICF. This additional context helps the model better understand and address the specific nuances of the task by using relevant domain information.

For the few-shot approach, two examples of input-output pairs are included in the prompt. This choice is driven by the need to balance between having sufficient examples and the availability of human-created examples. These examples are crafted by medical experts familiar with ICF categories and skilled in formulating natural questions for function monitoring and mental condition assessment. The examples remain consistent across different activities (second layer of ICF categories) and only vary by function (e.g., "mobility", "self-care" or "communication"). For instance, regardless of whether the conversation is about "Walking" or "Going up and down stairs", both belonging to the "mobility", the examples in the prompt are the same and may relate to different activities such as "Moving around using equipment" or "Moving around in different locations", also belonging to mobility. This approach is justified because the evaluation focuses on functions rather than specific activities, and also due to the limited number of human-created follow-up questions.

There are trade-offs between few-shot and zero-shot methods. More examples do not always lead to better conversation quality. For example, when experimenting with various shot numbers—five-shot, two-shot, and zero-shot—the conversations generated showed little difference between the two-shot and zero-shot methods. However, with five-shot prompting, there was a noticeable increase in off-topic responses. This may be due to the confusion introduced by having too many examples, which can result in less focused and more divergent outputs.

Additionally, for fine-tuned models, a basic instruction is used alongside zero-shot and few-shot prompts. The three types of instructions are classified based on the extent of information provided in the prompt:

1. **Basic Instruction:** Directs the model to play a role and ask follow-up questions about function-level or emotional feedback.
2. **Zero-Shot Prompt:** Includes the basic instruction and additional background knowledge from the ontology.
3. **Few-Shot Prompt:** Builds on the zero-shot prompt by including two specific examples.

Basic instructions are used for prompting baseline models (non-fine-tuned models) and fine-tuned models when no additional zero-shot or few-shot prompts are applied.

3.1.4 Experimental Set-up

This section introduces the experimental set-up for data generation, including data statistics, model introduction, parameter settings and evaluation.

Data Distribution In the current task, there are three primary categories: "mobility", "self-care", and "communication". For each activity (the second layer) within these categories, the model is tasked with generating 30-40 conversations, and for each conversation, the model is required to generate 6 Following-up questions of one type. That is, there are 12 FQs for each conversation. The distribution of activities across these categories is: Communication: 13 types; Self-care: 7 types; Mobility: 16 types. This data statistics is shown in 3.5 and the data distribution is illustrated in 3.1.

	Num of Conversations	Activity Types
Communication	390	13
Self-care	280	7
Mobility	640	16

Table 3.5: Total Number of Generated Conversations by GPT-3.5

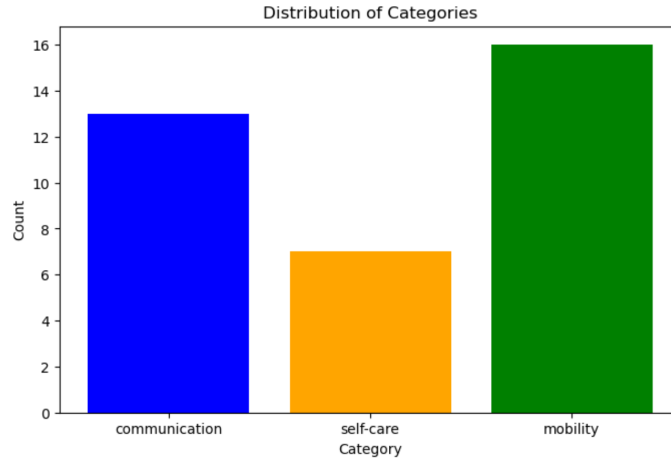


Figure 3.1: Number of Activity Types for Each Category

Model Introduction According to (Wang et al., 2021), GPT-3 leverages its advanced language generation capabilities to reduce both the time and cost associated with creating datasets, while maintaining quality comparable to human-generated data.

According to OpenAI (2024), GPT-3.5, an advancement over its predecessor GPT-3, incorporates several improvements that enhance its performance and versatility. It is built upon the same foundational architecture as GPT-3, utilizing a transformer-based model designed to process and generate human-like text (Ye et al., 2023). However, GPT-3.5 has been further refined through reinforcement learning with human feedback (RLHF), a technique that allows the model to learn from human evaluators who provide guidance on the quality and relevance of its outputs (Ye et al., 2023; Höglund and Khedri, 2023). This reinforcement learning process helps GPT-3.5 generate text that is not only more coherent and contextually accurate but also more nuanced and creative.

The model’s architecture involves processing input text by converting it into tokens, which are then analyzed through multiple transformer layers (Radford et al., 2019). These layers use attention mechanisms to determine the importance of each token in relation to others, allowing GPT-3.5 to generate responses that reflect a deep understanding of context and relevance. The attention mechanism assigns different weights to tokens based on their contextual significance, ensuring that the generated text is coherent and contextually aligned with the input prompt (Vaswani et al., 2023).

GPT-4, as the successor of GPT-3.5, offers even greater performance but is more expensive and not available for free (OpenAI, 2024). Compared to GPT-4, GPT-3.5 is a cost-effective option, which balancing cost and performance effectively. Therefore, GPT-3.5 is a practical choice for generating high-quality text across various applications and is applied in the current task.

Temperature Setting: The randomness of the generated output is controlled by the temperature parameter. In practice, adjusting the temperature allows for fine-tuning the model’s output to meet specific needs. Lower temperatures are preferred for tasks requiring precision and reliability, such as technical explanations or factual responses. Higher temperatures are suited for creative tasks, such as brainstorming or generating imaginative content. By carefully selecting the temperature setting, users can influence the style and quality of the generated text to align with their goals.

I experimented with three temperature settings: 0.5 (low), 1.0 (medium), and 1.5 (high). The results indicated that a temperature of 0.5 produced predictable and consistent conversations, 1.0 introduced some variety but remained somewhat repetitive, and 1.5 yielded the most creative and relevant responses. Consequently, I opted for a temperature setting of 1.5.

3.1.5 Prompt Evaluation

Prompt engineering is an iterative and often ad-hoc process, as there is no definitive theoretical framework explaining why certain prompt styles perform better with LLMs than others (Kaddour et al., 2023). Consequently, it is advantageous to experiment with various prompts and assess their effectiveness through multiple iterations. The evaluation of text generation typically involves two primary components: automatic evaluation and human evaluation (Chang et al., 2023).

In the context of prompt evaluation, human evaluation is crucial. It assesses whether the generated conversations are relevant, sensible, and creative. This approach allows for a nuanced understanding of the quality of outputs, which is essential for refining prompt strategies and ensuring that the generated text meets desired standards. Further details on evaluation methodologies will be discussed in the Evaluation Section.

3.2 Data Statics and Preprocess

The data statistics for the generated raw conversations are presented in Table 3.5. Upon inspection, two main issues were identified with the generated conversation data:

1. Many conversations contain nonsensical words or are irrelevant to the topic.
2. Some conversations exceed the maximum token limit.

Additionally, there are specific formatting requirements for preparing the training data for fine-tuning.

To address these issues, the generated data will undergo a two-step processing procedure:

1. **Filtering:** Conversations containing nonsensical words or irrelevant content will be removed. This filtering process will also be applied to the generated raw FQs based on the filtered conversation data.
2. **Formatting:** The remaining data will be formatted into a prompt structure suitable for fine-tuning the model.

3.2.1 Data Preprocess and Statics

In the first step of data processing, I validate the words within each conversation by assessing their validity based on various criteria, including recognition as English words, named entities, or specific suffixes. Libraries are leveraged such as spaCy for Named Entity Recognition (NER) and natural language processing, and NLTK for tokenization, lemmatization, and Part-of-Speech (POS) tagging. Additionally, a custom word list is predefined to address tokens that may be erroneously considered non-words but are, in fact, legitimate words.

To be more specific, the invalid words can be grouped into two types: non-English words such as Chinese or Korean, and words generated with random characters like "ampledapi", "pomelli", and "distintch". Both types are filtered based on the validity criteria mentioned above. However, some valid words like "sunscreen" and "smiley" may also be mistakenly considered as invalid. To handle this, I created a list of invalid words that occur more than once. This approach assumes that repeated invalid words are likely to be legitimate since real invalid words tend to be randomly produced. Although this method cannot identify valid words that appear only once in the conversations, it effectively captures most valid words that occur multiple times. The custom word list currently contains 31 entries.

To ensure coherence and relevance, I assess the semantic similarity between consecutive sentences and remove those with low coherence. Specifically, I use SpaCy with the

”en_core_web_md” model, which provides word and sentence embeddings for calculating similarity scores. If the similarity score between two consecutive sentences is below 0.5, they are deemed to lack coherence. In practice, most sentence pairs fall below this threshold, indicating that none are considered sufficiently similar to be coherent. Although a lower threshold might allow for some level of coherence, the method proved ineffective. In some cases, pairs of sentences that seemed more semantically related received lower similarity scores than pairs that appeared less related. As a result, I decided not to rely on this method for coherence evaluation.

If any token in a conversation is identified as a non-word, the entire conversation or follow-up questions are discarded. Furthermore, to prepare the data for FQs, I delete the last utterance in conversations if spoken by caretakers. For each activity, I requested a total of 30-40 conversations, resulting in varying numbers of conversations across categories. The filtered data is summarized in Table 3.6, where 64 conversations have been filtered for Communication, 96 for Mobility and 38 for Self-care.

For the follow-up questions (FQs), the same filtering pipeline was applied to eliminate invalid texts. Initially, six raw FQs were generated for each conversation, one for each FQ type, resulting in a total of 12 FQs per conversation. After the filtering process, only the first FQ from each type was retained, as the quality of the first FQ was found to be more consistent upon inspection. Consequently, each conversation is now paired with two FQs—one focusing on the functional level and the other on emotional feedback. Thus, for each conversation, only two FQs are retained, one from each FQ type.

Category	Activity types	Conversation Num	Filtered Conversation Num
Communication	13	326	64
Mobility	16	544	96
Self-Care	7	242	38

Table 3.6: Filtered Conversation Statistics

After the data preprocessing, the conversations are split into training data, validation data and test data almost as 0.7, 0.125, 0.125, shown as Table 3.7.

Category	Train	Validation	Test
Communication	222	41	63
Mobility	375	72	97
Self-Care	166	33	43

Table 3.7: Number of Conversation over Train, Validation and Test for Three Categories

3.2.2 Training Data Statics and Format

A total of 222 conversations and 444 corresponding follow-up questions (FQs) were used for training. The distribution of conversations across categories is illustrated in Figure 3.2. Mobility has the highest proportion of training conversations at 44.4%, followed by communication at 36.1%, and self-care at 19.4%. For a more detailed breakdown of conversations by activity, refer to Figures 3.3 (Communication), Figure 3.4 (Self-care), Figure 3.5 (Mobility). These figures demonstrate that conversations are evenly distributed across activities within each category. The specific number of conversations for each activity is detailed in Table 3.8. FQs for training share the same distribution with conversation distribution.

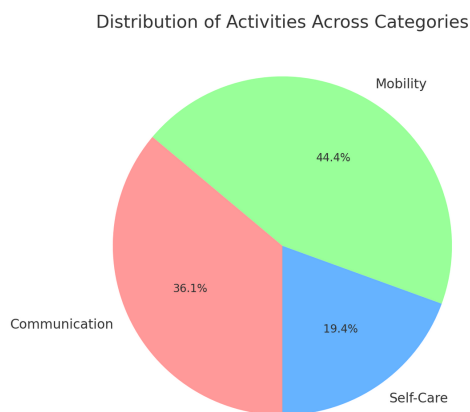


Figure 3.2: Distribution of Activities across Categories

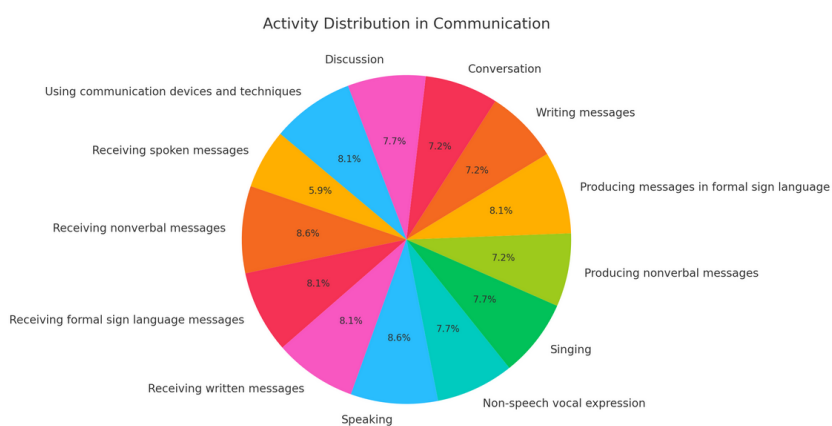


Figure 3.3: Activity Distribution in Communication

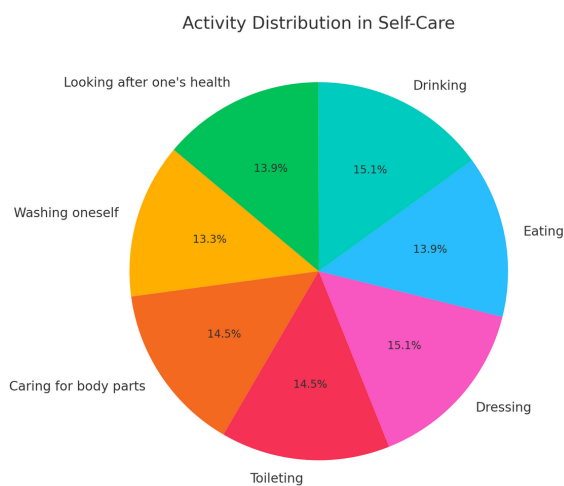


Figure 3.4: Activity Distribution in Self-care

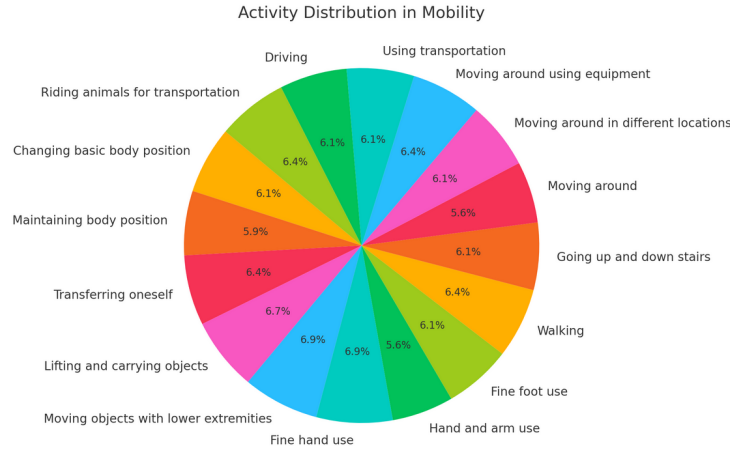


Figure 3.5: Activity Distribution in Mobility

After generating the training data, it must be prepared in a format suitable for fine-tuning. The prompt format used for fine-tuning Llama3 differs from the format used during data generation.

Llama3’s prompt format is a template containing placeholders for task descriptions and example data (Zhao et al., 2021). Specifically, it consists of four key fields (Honovich et al., 2022):

1. **Instruction:** A description of the task, typically generated as a generic template.
2. **Input Argument:** This instantiates the instruction, providing a specific example of the task.
3. **Output Space Constraints:** Details to restrict the output space. This field is primarily relevant to classification tasks and will not be used for the current task.
4. **Textual Output:** A reflection of a correct execution of the instruction based on the given input arguments and output constraints.

For Llama3, the prompt format is structured as follows:

```
prompt = """Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.
```

```
### Instruction: {}
```

```
### Input: {}
```

```
### Response: {}"""
```

The *Instruction* will be filled in with the basic instruction, requiring the model to play roles and ask follow-up questions about function-level or emotional feedback with the format requirement. The *Input* will be filled in with the conversation history and the *Response* will be filled in with the FQs. Taking the activity of “Communicating with receiving spoken messages” with the FQs on function level as an example,

```
prompt = """Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.
```

```
### Instruction:
```

You will be given a conversation history between a caretaker and a patient about one activity. You need to ask follow-up questions to evoke answers informing about the function level. The questions you ask should have around 2 to 6 utterances, and each utterance should be completed and have less than 20 tokens.

The format is as follows:

C: utterance (asking follow-up questions)

P: utterance (respond naturally)

C: utterance (asking follow-up questions)

P: utterance (respond naturally)

Input:

{C: How was your day? Did you enjoy any interesting conversations today?

P: Yes, I had a lovely chat with one of the nurses this morning.

C: That's great to hear! What did you two talk about?

P: Oh, we talked about gardening and the beautiful flowers in the garden here.

C: It sounds like a pleasant conversation. I'm glad you had a nice chat.

P: Thank you. It's nice to have friendly conversations to brighten up the day.}

Response:

{C: It's wonderful to hear about your conversation. How do you generally feel about communicating and understanding spoken messages? Any particular difficulties?

P: Well, communicating and understanding spoken words is slightly challenging, but I manage to grasp most of what's being said with some effort.

C: I see. Has it always been a mild challenge or is it something that has worsened over time when it comes to following spoken conversations?

P: It's been a slight difficulty for a while now, especially when there's too much background noise, making it a bit harder to comprehend sometimes. }"""

The joined instruction and input-out training pairs will be used for the prompt of Llama3.

Category	Activity	Num of Conversations
Communication	Communicating with receiving spoken messages	13
	Communicating with receiving nonverbal messages	19
	Communicating with receiving formal sign language messages	18
	Communicating with receiving written messages	18
	Speaking	19
	Non-speech vocal expression	17
	Singing	17
	Producing nonverbal messages	16
	Producing messages in formal sign language	18
	Writing messages	16
	Conversation	16
	Discussion	17
	Using communication devices and techniques	18
	Self-Care	Washing oneself
Caring for body parts		24
Toileting		24
Dressing		25
Eating		23
Drinking		25
Looking after one's health		23
Mobility	Changing basic body position	23
	Maintaining body position	22
	Transferring oneself	24
	Lifting and carrying objects	25
	Moving objects with lower extremities	26
	Fine hand use	26
	Hand and arm use	21
	Fine foot use	23
	Walking	24
	Going up and down stairs	23
	Moving around	21
	Moving around in different locations	23
	Moving around using equipment	24
	Using transportation	23
	Driving	23
Riding animals for transportation	24	

Table 3.8: Num of training conversations of each Activity

3.2.3 Validation and Test Data

Human conversations along with the FQs are required for the project for validation and test. However, due to privacy protection of medical data, these kinds of user conversations are not available along with the following-up questions. We therefore asked three experts with experience in healthcare from the A-PROOF team to create those data, who are all familiar with the ICF framework. Considering the time limitation and the primary purpose of the current task, they will be given the generated conversations and required to create the FQs with their professions determining how to ask the following-up questions for monitoring the function level and emotional conditions.

They will also given a guideline for the data creation, which is similar to the prompt,

1. You will be provided with a conversation about an activity between a care-taker (C) and an elderly patient (P). The topics of each conversation should fall under one subcategory of ICF activities and participation.
2. Your task is to continue this conversation by asking follow-up questions, playing the roles of both the care-taker and the elderly patient.
3. Your goal is to ask two types of follow-up questions (FQ). The first one is FQs on function level, aiming to evoke answers that reflect the patient’s functional abilities. The second one is FQs on emotional feedback, seeking emotional feedback regarding the target activities in the conversation (positive or negative feelings).
4. Two types of questions are independent form each other. Emotional feedback questions should be based on the given conversation, not on the function level questions.

Both the Validation data and Test data of conversations were provided to medical experts for the creation of natural follow-up questions (FQs). However, due to time constraints, the validation dataset is not filled and the test dataset is partially complete. Also, some of the provided conversations are regarded as exotic and thus be removed from the test dataset. The remaining data statistics is shown in Table 3.9. This table also includes the statistics for both types of FQs: those focused on the functional level and those addressing emotional feedback.

Category	Total Conversations	Total FQ on Function Level	Total FQ on Feedback
Communication	36	36	35
Mobility	52	52	32
Self-Care	25	25	25

Table 3.9: Reference amount of Conversations and FQs

The data distribution is similar to training data. shown as figures 3.6 and 3.7.

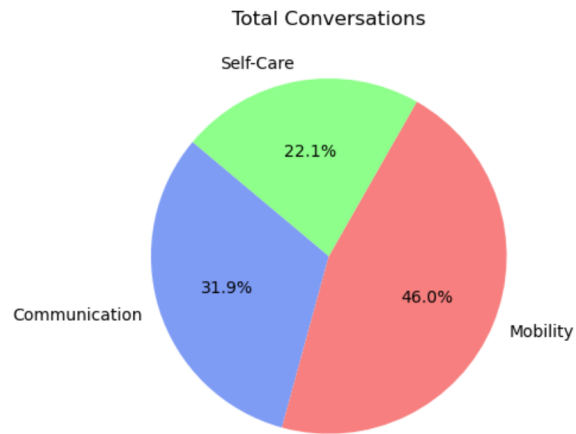


Figure 3.6: Conversation Distribution of Test

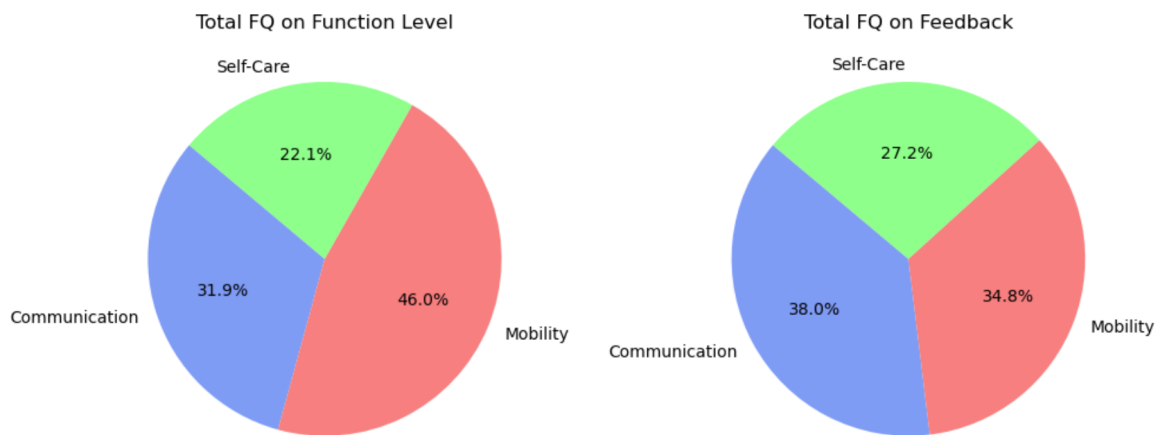


Figure 3.7: FQs Distribution of Test

Also, more detailed activity-based reference statistics is provides in Table 3.10, showing the statistics of conversations, FQ on function level (Func-FQ), and FQ on emotional feedback (Emo-FQ) used in test data.

Category	Subcategory	Convers	Func-FQ	Emo-FQ	
Communication	Communicating with receiving spoken messages	4	4	4	
	Communicating with receiving nonverbal messages	3	3	2	
	Communicating with receiving formal sign language messages	2	2	2	
	Communicating with receiving written messages	3	3	3	
	Speaking	3	3	3	
	Non-speech vocal expression	2	2	2	
	Singing	3	3	3	
	Producing nonverbal messages	3	3	3	
	Producing messages in formal sign language	2	2	2	
	Writing messages	2	2	2	
	Conversation	3	3	3	
	Discussion	3	3	3	
	Using communication devices and techniques	3	3	3	
	Mobility	Changing basic body position	6	6	6
		Maintaining body position	6	6	3
Transferring oneself		5	5	2	
Lifting and carrying objects		4	4	2	
Moving objects with lower extremities		4	4	2	
Fine hand use		4	4	2	
Hand and arm use		4	4	2	
Fine foot use		2	2	0	
Walking		4	4	2	
Going up and down stairs		2	2	2	
Moving around		2	2	2	
Moving around in different locations		2	2	2	
Moving around using equipment		3	3	1	
Using transportation		2	2	2	
Driving		2	2	2	
Self-Care	Washing oneself	6	6	6	
	Caring for body parts	6	6	6	
	Toileting	5	5	5	
	Dressing	2	2	2	
	Eating	2	2	2	
	Drinking	2	2	2	
	Looking after one's health	2	2	2	

Table 3.10: Detailed Breakdown of Conversations, Function Level, and Feedback Quality

3.3 Experimental Set-up

This section first introduces the generative model chosen for the project and then the hyperparameter tuning. The last sub-section introduces the fine-tuning approach for optimizing the models.

3.3.1 Model Introduction: LLaMA3

According to Touvron et al. (2023), LLaMA (Large Language Model Meta AI) is a series of foundational language models developed by Meta AI, with versions ranging from 7 billion to 65 billion parameters. These models are trained on publicly available datasets and have achieved state-of-the-art performance in various benchmarks. Notably, LLaMA-13B has outperformed GPT-3 on many benchmarks, despite being significantly smaller. Meta AI has released these models to the research community to promote accessibility and encourage further study of large language models (LLMs).

Several factors influenced the choice of LLaMA-3 for this task. First, LLaMA-3 strikes a favorable balance between size and performance, making it a practical choice for various applications. Additionally, LLaMA-3 is designed to be cost-effective at inference time. This characteristic is crucial for deploying language models at scale, as it ensures that the model is not only efficient to train but also economical and fast in generating text.

Moreover, LLaMA-3 benefits from minimal instruction fine-tuning, which can significantly enhance its performance for specific tasks. This adaptability indicates that LLaMA-3 can be effectively customized to meet particular requirements, making it a suitable choice for optimizing performance on targeted tasks.

3.3.2 Model Parameter Tuning

The performance of models can be influenced by several hyperparameters, including learning rate and batch size. It is crucial to conduct multiple experiments to fine-tune these parameters on the validation dataset to optimize model performance.

- **Learning Rate** Learning Rate controls the size of steps that the optimization algorithm takes when adjusting the model's weight to minimize the loss function. When the learning rate is set overly large, the model may take large steps in the direction of gradient, resulting in overshooting the optimal point and leading to increased loss. When it's set too small, the model may take too small steps, resulting in extremely slow process to converge. A well-chosen learning rate helps the model converge efficiently to a good solution.
- **Batch Size** This is the number of training examples used in one forward and backward pass. Larger batch sizes can stabilize training and make better use of computational resources but require more memory. Smaller batch size can make the training process less stable, but it may help the model escape local minima.

A common heuristic for selecting the learning rate is that the optimal value is often close to the largest rate that does not cause divergence in the training process (Bengio, 2012a). To tune this hyperparameter, one effective approach is to start with a relatively large learning rate and, if divergence occurs, reduce the learning rate by a factor of three, and repeat this process until the training stabilizes (Bengio, 2012b,a).

In practice, models are tested with a range of learning rates on the validation dataset to monitor performance. Iterative adjustments are made based on validation results, aiming to improve metrics such as accuracy or F1 score. The final set of hyperparameters, which yields

the best performance on the validation dataset, is then used for the final test to generate inferences.

However, due to time constraints, I ultimately opted for the default learning rate of $2e-4$. This decision was influenced by two key factors: Firstly, there was no available validation dataset within the time limit. Secondly, default learning rates provided with pre-trained models or specific frameworks (e.g., Transformers in Hugging Face) are typically based on extensive experimentation and tuning by the original developers. These defaults are generally well-optimized for a broad range of tasks, ensuring reliable and consistent performance. Using these default settings allows for good performance while saving time and ensuring more consistent results.

3.3.3 Optimization Approaches

Inspired by (Chen et al., 2024; Brown et al., 2020), there are three approaches for improving the performance of LLMs: Fine-Tuning, Few-shot prompting, and Zero-shot prompting.

Fine-tuning is a method used to adapt a pre-trained language model to perform well on a specific task or within a particular domain. This process involves taking a model that has already been trained on a large, general dataset and further training it on a smaller, task-specific dataset. The goal of fine-tuning is to leverage the broad knowledge the model has acquired during its initial training while tailoring it to the nuances of the specific application.

Fine-tuning is highly effective and has been shown to achieve superior performance across a range of benchmarks compared to training models from scratch. By starting with a pre-trained model, fine-tuning can significantly reduce the amount of data and computational resources required compared to training a model from the ground up. This approach allows the model to benefit from the rich features learned from large-scale data, which can then be refined with targeted, domain-specific data.

However, despite its advantages, fine-tuning comes with several drawbacks. It requires a considerable amount of high-quality, task-specific data and substantial memory to handle models with billions of parameters. Although the use of generated training data can help reduce data requirements, fine-tuning still involves challenges related to computational efficiency. Specifically, it necessitates storing and managing individual copies of the fine-tuned models for each system or task, which can be both memory-intensive and computationally inefficient (Kaddour et al., 2023).

This method, while powerful, underscores the need for efficient management strategies to optimize resource use and maximize the benefits of fine-tuning in practical applications.

LoRA (Low-Rank Adaptation) adapters (Hu et al., 2021) are used to address the challenges associated with fine-tuning large language models. LoRA improves the fine-tuning process by introducing low-rank matrices into specific layers, typically within the attention mechanisms, instead of updating all the model parameters. These low-rank matrices, which have significantly fewer parameters compared to the full model, greatly reduce computational overhead.

By saving only the small adapter parameters rather than the entire fine-tuned model, LoRA minimizes memory usage. The base model remains unchanged, and only the adapter parameters are adjusted to fine-tune the model for different tasks. This approach allows for the storage of multiple LoRA adapters without duplicating the entire model, thereby optimizing scalability and reducing storage requirements. Additionally, LoRA enables quick task switching by loading different adapter sets without the need to retrain or reload large models. As a result, LoRA adapters offer a highly efficient and flexible solution for deploying large language models. They reduce storage and memory needs, lower computational costs, and maintain performance comparable to full fine-tuning. In the current task, the final fine-tuned LLaMa3 model will be saved as LoRA adapters based on different datasets, which will be loaded for inference as needed.

Few-shot and Zero-shot prompting are techniques used in natural language processing to leverage pre-trained models for performing specific tasks with minimal or no task-specific training. They have been introduced in more details in Section. Data Collection for prompt engineering.

To recap, Zero-shot prompting involves providing a model with a clear and detailed instruction or prompt about the task without any specific examples. The model relies on its broad pre-existing knowledge from extensive initial training to generate responses. On the other hand, Few-shot prompting, specifically two-shot prompting in the current project, includes providing two example input-output pairs within the prompt alongside the task instruction. These examples offer additional context, helping the model better understand and execute the task.

To summarize, we will compare two main systems and five different models. The two systems are:

- Fine-tuned System
- Non-fine-tuned System

The five models to be compared are:

1. **FT Zero** The fine-tuned LLaMA3 model trained on zero-shot prompted data using GPT-3.5.
2. **FT Few**: The fine-tuned LLaMA3 model trained on few-shot prompted data using GPT-3.5.
3. **Pre Zero**: The pre-trained LLaMA3 model using zero-shot prompting.
4. **Pre Few**: The pre-trained LLaMA3 model using few-shot prompting.
5. **Baseline**: The pre-trained LLaMA3 model using basic instruction without zero-shot or few-shot prompting.

3.4 Evaluation

The evaluation of question generation encompasses two main components: automatic evaluation and human evaluation (Chang et al., 2023). Additionally, the output data and references require post-processing for effective evaluation.

3.4.1 Data Post-process

Data post-processing prepares the data for evaluation by focusing on relevant utterances. Specifically, the process involves extracting follow-up questions (FQs) from the model-generated outputs. Since only the FQs, which are the utterances from care-takers, are needed for evaluation, all other utterances are excluded. For each conversation, there is typically one FQ of each type, while multiple references may exist for a single conversation. Below, we outline the procedure for using reference sets to evaluate inferences.

Consider the following example conversation (3.11):

Speaker	Dialogue
C	Have you walked anywhere yesterday?
P	I walked to the cinema yesterday.

Table 3.11: Example Dialogue

From this conversation, the model might generate FQs as follows (3.12):

Speaker	FQs
C	How far is the cinema from your place, and how long did it take you to walk there?
P	The cinema is about a mile away, and it took me around 20 minutes to get there.
C	Do you need to have a break during the walking?
P	Actually, I didn't pay attention. We walked there while chatting.

Table 3.12: Model-generated FQs on Function Level

To prepare for evaluation, the FQs are extracted as follows (3.13):

Inference ID	Functional Question (FQ)
1	(C:) How far is the cinema from your place, and how long did it take you to walk there?
2	(C:) Do you need to have a break during the walking?

Table 3.13: Extracted FQ Inferences

Assuming there are two reference sets for comparison, the first set contains four utterances, and the second set contains two utterances (3.14):

Reference Set	Speaker	Reference FQs
Set 1	C	Did you need any assistance when you walked?
	P	No, I did that alone.
	C	You didn't use a walker, either?
	P	Oh, I used that. I couldn't walk by myself.
Set 2	C	How did you feel during and after the walk? Were you comfortable, or did you experience any pain or fatigue?
	P	I felt fine during the walk, but my legs were a bit sore afterward. It wasn't too bad, though.

Table 3.14: Reference FQs

The extracted FQs from care-takers will also be reviewed (3.15):

	Reference Set 1	Reference Set 2
(1)	(C:) Did you need any assistance when you walked?	(C:) How did you feel during and after the walk? Were you comfortable, or did you experience any pain or fatigue?
(2)	(C:) You didn't use a walker, either?	<i>No second reference question</i>

Table 3.15: Extracted FQ Reference Sets

Each inference is evaluated by comparing it with corresponding reference questions from the reference sets. The evaluation process involves calculating scores by either averaging or selecting the maximum score among the available references. For instance (3.16):

Inference	Reference Set 1	Reference Set 2
1. How far is the cinema from your place, and how long did it take you to walk there?	(1) Did you need any assistance when you walked?	(2) How did you feel during and after the walk? Were you comfortable, or did you experience any pain or fatigue?
2. Do you need to have a break during the walking?	(2) You didn't use a walker, either?	<i>No second reference question</i>

Table 3.16: Evaluation of Inferences against Reference Questions

To evaluate the inferences, each one is compared with its corresponding reference questions. If multiple references are available for a given inference, evaluation metrics can either average the scores across all references or select the maximum score among them. Each inference FQ is assigned a score based on these metrics. The overall evaluation score for each function or category is then determined by averaging the scores of all FQ inferences associated with that particular function (categories of mobility, communication or self-care).

3.4.2 Automatic evaluation

The performance of question generation is usually evaluated by the following three traditional automatic metrics (referred as *traditional metrics*) employed in machine translation evaluation. It includes BLEU, METEOR, ROUGE.

BLEU (Bilingual Evaluation Understudy) is a precision-based metric (Papineni et al., 2002) that assesses how well a candidate output matches a set of reference texts. It calculates the percentage of n-grams in the candidate text that overlap with the references, with a penalty applied for overly short sentences.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004) focuses on recall and evaluates overlap using various n-gram and word-based measures. ROUGE-N measures recall by comparing n-grams in the candidate text to those in the reference summaries and specifically, ROUGE-L evaluates recall based on the Longest Common Subsequence (LCS) statistics, assessing how much of the reference text’s word order is captured in the candidate.

METEOR (Metric for Evaluation of Translation with Explicit Ordering) (Denkowski and Lavie, 2014) is a recall-oriented metric that computes sentence-level similarity scores between the generated output and reference texts. It incorporates factors such as synonymy, stemming, word order, and paraphrasing into its evaluation.

Taking the following reference and inference candidate as an example and the calculation is shown as the Table 3.17:

Reference: The cat sat on the mat.

Candidate: The cat is sitting on the mat.

Metric	Reference	Candidate	Match	Score
BLEU (2-gram)	the cat, cat sat, sat on, on the, the mat	the cat, cat is, is sit- ting, sitting on, on the, the mat	the cat, on the, the mat	$3/5 = 0.600$
ROUGE-1	the, cat, sat, on, the, mat	the, cat, is, sitting, on, the, mat	the, cat, on, the, mat	$5/6 = 0.833$
ROUGE-L	The cat sat on the mat	The cat is sitting on the mat	LCS: The cat [] on the mat	LCS Length: 8 Reference Length: 9 Candidate Length: 7 $8/9 = 0.889$
METEOR	the, cat, sat, on, the, mat	the, cat, is, sitting, on, the, mat	Matches: the, cat, on, the, mat	0.750

Table 3.17: Evaluation of metrics with example data.

In the current project, BLEU is utilized with 4-grams, ROUGE uses one-gram, bi-gram, and ROUGE-L, while METEOR is also employed. Although these metrics are standard for automatic language evaluation, their effectiveness can be limited. Word-overlap metrics may struggle to fully capture the quality of responses in dialogue, where multiple appropriate responses can exist even for a given conversation and limited types of follow-up questions (Zhao et al., 2017; Yeh et al., 2021). To address these limitations, an advanced method such as BERTScore may be used for more nuanced evaluation.

BERTScore is a metric that evaluates the quality of generated text by computing the F1 score based on token embeddings from a pre-trained language model, such as BERT

(Bidirectional Encoder Representations from Transformers) (Zhang et al., 2020). Unlike traditional metrics like BLEU or ROUGE, which rely on n-gram overlap, BERTScore captures semantic similarity by leveraging contextual embeddings. This approach allows for a more nuanced comparison of the candidate and reference sentences, as it considers the contextual meaning of words rather than just their surface forms.

BERTScore calculates precision, recall, and F1-score by comparing how well the tokens in the candidate sentence match the tokens in the reference sentence based on their embeddings. The scores are computed as follows:

- *Precision* measures the accuracy of the positive predictions made by the model. It is the ratio of correctly predicted positive observations to the total predicted positives. In the context of BERTScore, precision indicates how many of the predicted tokens in the candidate sentence have a close semantic match with the tokens in the reference sentence.
- *Recall* measures the model’s ability to identify all relevant cases within the data. It is the ratio of correctly predicted positive observations to all the observations in the actual class. In BERTScore, recall indicates how well the tokens in the reference sentence are matched by those in the candidate sentence.
- *F1-score* is the harmonic mean of precision and recall, providing a balanced metric that accounts for both accuracy and completeness. The F1-score is particularly useful when there is a need to balance precision and recall, especially in cases where there is an imbalance in the data or when the cost of false positives and false negatives is similar.

BERTScore provides a comprehensive measure of the quality of the generated text by considering the semantic and contextual relationships between words, offering an advanced alternative to traditional n-gram-based evaluation metrics.

3.4.3 Human evaluation

Automatic metrics have been found to have limited effectiveness in evaluating language generation systems, which has led researchers to primarily rely on human evaluation (Zhang et al., 2021). Human evaluation provides invaluable and nuanced insights into the quality of generated content. However, it is both costly and time-consuming because setting up and conducting such evaluations requires meticulous planning to accurately reflect real-world conditions. As a result, human evaluation is typically reserved for the final stages of assessment.

During the development phase, automatic evaluation metrics play a critical role. They are essential for optimizing model design and selecting appropriate hyperparameters, providing quick and consistent feedback that guides the iterative process of model improvement (Liu et al., 2017; Deriu et al., 2020).

This method is particularly useful in prompt engineering, where different prompt designs need to be evaluated rapidly to identify the most effective approaches for generating high-quality responses.

Chapter 4

Results and Analysis

This chapter involves presenting the results of experiments and analyzing the performance of two systems and five models that generate follow-up questions based on input conversations. Two systems include (1) Fine-tuned System (2) Non-fine-tuned with prompt approach System. Five models include (1) **FT Zero** The fine-tuned LLaMA3 model trained on zero-shot prompted data using GPT-3.5. (2) **FT Few**: The fine-tuned LLaMA3 model trained on few-shot prompted data using GPT-3.5. (3) **Pre Zero**: The pre-trained LLaMA3 model using zero-shot prompting. (4) **Pre Few**: The pre-trained LLaMA3 model using few-shot prompting. (5) **Baseline**: The pre-trained LLaMA3 model using basic instruction without zero-shot or few-shot prompting. The evaluation metrics has been divided into traditional metrics of BLEU, ROUGE and METEOR as well as more advanced metrics of BERT-scores.

Results and analysis will be divided into three groups for comparison:

1. **System-wise Comparison**: this comparison is to analyze the overall performance of each system on each class along with within-system comparison.
2. **Category-wise Comparison**: this comparison is to analyze the results of categories across two questions types.
3. **FQ type-wise Comparison**: this comparison is to analyze the results of two FQ types across three categories.

4.1 Results

This section will present the results of four models over all categories with two types of FQ types. To make the illustration clearer, I refer individual category with individual question type as class. Therefore, there are six classes will be compares: (1) *S-Care Emo*: Self-care with FQs on emotional feedback (2) *S-Care Func*: Self-care with FQs on functional level (3) *Com Emo*: Communication with FQs on emotional feedback (4) *Com Func*: Communication with FQs on function level (5) *Mob Emo*: Mobility with FQs on emotional feedback (6) *Mob Func*: Mobility with FQs on function level.

The results of baseline is presented in Table 4.1. The results of four models are presented with traditional metrics in Table 4.2) and with BERT-scores Table 4.3 respectively. In each table, the highest scores of each class have been bolded.

Within traditional metrics, **FT Few** (fine-tuned LLaMA trained on few-shot prompted training data) consistently delivered the highest performance across most traditional metrics as indicated by the bold results in Table 4.2, particularly in the class of *S-Care Emo* (Self-care with FQs on emotional feedback), gaining the highest ROUGE-L of 0.2128 and METEOR of 0.2711; in the class of *Com Emo* (Communication with FQs on emotional feedback) with all the highest traditional metrics: BLEU4: 0.0330, ROUGE1: 0.2730, ROUGE2: 0.0834,

ROUGE-L: 0.2301 and METEOR: 0.2951; as well as in the class of *Mob Emo* (Mobility with FQs on emotional feedback), also gaining all the highest traditional metric scores. **FT Zero** (fine-tuned LLaMA with zero-shot prompting) followed by, which also had high performance across lots of traditional metrics in the class of *S-Care Func* (Self-care with FQs on functional level), *Com Func* (Communication with FQs on functional level) and *Mob Func* (Mobility with FQs on function level). **Pre Few** and **Pre Zero** exhibit relatively poor performance, with **Pre Zero** being especially underwhelming. Neither consistently achieves the highest scores across any class when evaluated using any traditional metrics.

Class	BLEU4	ROUGE1	ROUGE2	ROUGE-L	METEOR	Precision	Recall	F1
S-Care Emo	0.0000	0.0000	0.0000	0.0000	0.0000	0.7414	0.7246	0.7328
S-Care Func	0.0000	0.0000	0.0000	0.0000	0.0000	0.6735	0.6594	0.6663
Com Emo	0.0000	0.0000	0.0000	0.0000	0.0000	0.7737	0.7579	0.7656
Com Func	0.0000	0.0000	0.0000	0.0000	0.0003	0.6867	0.6762	0.6813
Mob Emo	0.0000	0.0000	0.0000	0.0000	0.0000	0.6504	0.6398	0.6450
Mob Func	0.0000	0.0006	0.0000	0.0006	0.0004	0.6144	0.6023	0.6082

Table 4.1: Baseline Average Scores Across All Class

Table 4.2: Average Scores for Traditional Metrics per Class

Class	Metric	FT Zero	FT Few	Pre Zero	Pre Few
S-Care Emo	BLEU4	0.0268	0.0456	0.0214	0.0508
	ROUGE1	0.2134	0.2362	0.1932	0.2445
	ROUGE2	0.0504	0.0801	0.0386	0.0890
	ROUGE-L	0.1897	0.2128	0.1737	0.2115
	METEOR	0.2317	0.2711	0.2008	0.2448
S-Care Func	BLEU4	0.0170	0.0130	0.0145	0.0240
	ROUGE1	0.1651	0.1483	0.1601	0.1615
	ROUGE2	0.0311	0.0225	0.0324	0.0343
	ROUGE-L	0.1459	0.1248	0.1406	0.1416
	METEOR	0.1896	0.1710	0.1757	0.1663
Com Emo	BLEU4	0.0310	0.0330	0.0225	0.0214
	ROUGE1	0.2493	0.2730	0.2111	0.2036
	ROUGE2	0.0799	0.0834	0.045	0.0429
	ROUGE-L	0.2201	0.2301	0.183	0.1653
	METEOR	0.2724	0.2951	0.2006	0.2211
Com Func	BLEU4	0.0226	0.0160	0.0272	0.0200
	ROUGE1	0.2096	0.1899	0.1932	0.1917
	ROUGE2	0.0459	0.0327	0.0352	0.0289
	ROUGE-L	0.1805	0.1676	0.17	0.1690
	METEOR	0.2128	0.1937	0.1943	0.1848
Mob Emo	BLEU4	0.0299	0.0483	0.0215	0.0281
	ROUGE1	0.2339	0.2524	0.2105	0.2153
	ROUGE2	0.0846	0.0994	0.0493	0.0617
	ROUGE-L	0.2146	0.2245	0.1858	0.1934
	METEOR	0.2426	0.2475	0.1896	0.2121

Continued on next page

Class	Metric	FT Zero	FT Few	Pre Zero	Pre Few
Mob Func	BLEU4	0.0216	0.0184	0.0156	0.0190
	ROUGE1	0.1792	0.1780	0.1599	0.1737
	ROUGE2	0.0392	0.0370	0.0351	0.0321
	ROUGE-L	0.1564	0.1547	0.1433	0.1540
	METEOR	0.1815	0.1852	0.1575	0.1632

Table 4.3: Average BERT Scores Across All Systems

Class	Metric	FT Zero	FT Few	Pre Zero	Pre Few
S-Care Emo	Precision	0.8746	0.8750	0.8638	0.8739
	Recall	0.8837	0.8848	0.8735	0.8772
	F1	0.8790	0.8798	0.8685	0.8754
S-Care Func	Precision	0.8594	0.8578	0.8601	0.8584
	Recall	0.8714	0.8704	0.8679	0.8680
	F1	0.8653	0.8640	0.8639	0.8630
Com Emo	Precision	0.8806	0.8800	0.8724	0.8642
	Recall	0.8855	0.8879	0.873	0.8716
	F1	0.8830	0.8838	0.8726	0.8678
Com Func	Precision	0.8693	0.8662	0.8689	0.8646
	Recall	0.8770	0.8750	0.8711	0.8724
	F1	0.8731	0.8705	0.8699	0.8684
Mob Emo	Precision	0.8746	0.8706	0.8668	0.8691
	Recall	0.8788	0.8779	0.8705	0.8730
	F1	0.8766	0.8741	0.8685	0.8710
Mob Func	Precision	0.8641	0.8611	0.8599	0.8653
	Recall	0.8719	0.8692	0.862	0.8691
	F1	0.8678	0.8651	0.8608	0.8678

For the BERT-Scores, which measures the semantic similarity between the generated and reference questions **FT Zero** and **Pre Few** still showed competitive performance, indicated by the bold parts in Table 4.1. Particularly, **FT Zero** tended to have slightly higher precision, recall and F1 scores across most class. Taking F1-score for presentation since F1-score is a balanced indicator of recall and precision, **FT Zero** has gained the highest F1-score in 5 classes *S-Care Func*: 0.8653, *Com Func*: 0.8731, *Mob Emo*: 0.8766, *Mob Func*: 0.8678. **FT Few**, on the other hand, gained the highest F1-score in 2 classes of *S-Care Emo*: 0.8798, *Com Emo*:0.8838. Additionally, **Pre Few** has gained highest score in *Mob Func*. Similar to the case with traditional metrics, **Pre Zero** has lowest performance across precision, recall and F1-scores.

The Baseline model shown in Table 4.1 performed poorly with both traditional metrics and BERT scores. Specifically, with traditional metrics, nearly all class scores are close to zero. The BERT scores for the Baseline system are also generally low, with the highest score remaining below 0.80. This is relatively poor, especially compared to other systems, all of which achieve higher scores.

As outlined in the previous section, the BERT-F1 score, which combines the advanced semantic similarity captured by BERT with the balanced precision and recall of the F1-score, will be utilized for the following comparison.

4.2 System-wise Comparison

This section compares two systems, the fine-tuning approach and the prompting approach, against both a baseline model and each other. The fine-tuning system includes two models: FT Zero (fine-tuned on zero-shot prompted data) and FT Few (fine-tuned on few-shot prompted data). The prompting system, on the other hand, consists of Pre Zero (non-fine-tuned models using zero-shot prompts) and Pre Few (non-fine-tuned models using few-shot prompts).

According to the result section above, FT Few and FT Zero consistently delivered the highest performance across most classes whereas Pre Few and Pre Zero show poor performance, particularly the latter gaining no highest score among all the classes. It indicates that the fine-tuning system outperforms the prompting system. It can be further illustrated by Figure 4.1 where the fine-tuning system is represented by the red and blue lines at the top and the prompting system is represented by the two lines at the bottom.

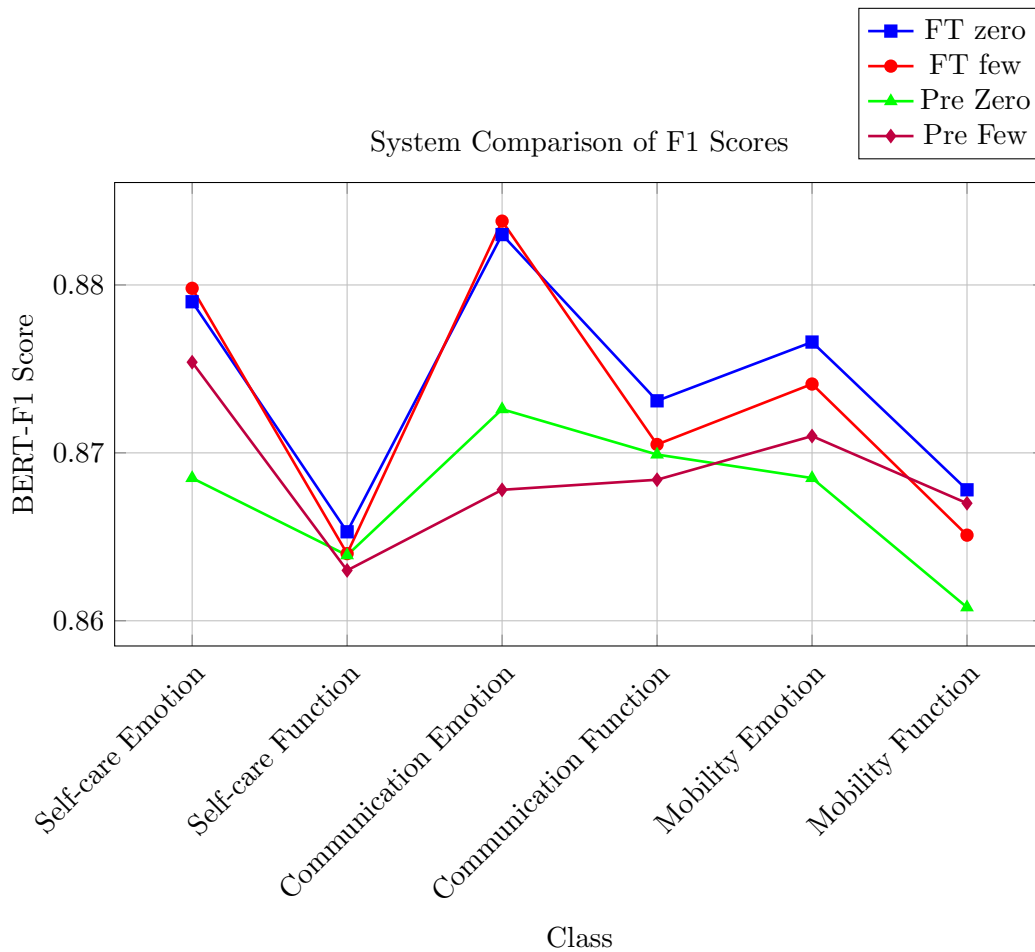


Figure 4.1: System Comparison of F1 Scores

In contrast, the prompting system demonstrates lower performance in F1-scores. While few-shot prompting without additional fine-tuning offers slight performance improvements, it still lags behind the effectiveness achieved through comprehensive fine-tuning.

However, when compared to the Baseline model in Table 4.1, whose highest score (0.7656) falls below this threshold (0.86) of all four models, all four models show higher performance and demonstrate substantial improvements over the baseline. This indicates that both fine-tuning and prompting approaches are effective for enhancing model performance. Nonetheless, the fine-tuning approach remains the key differentiator, as it enhances both the quality

and semantic relevance of the generated follow-up questions, outperforming both prompting models.

Additionally, within each system, there are variations in performance among the models. Among the fine-tuned models, FT Zero, which utilizes zero-shot prompted training, shows the higher performance on most class than FT Few. It particularly excels in generating questions related to the category of mobility and producing questions related to all FQ types on function level. In fact, FT Zero achieves the highest scores across mobility on both question types as well as achieves the highest scores across all categories in terms of the question type of function.

Within the prompting system, Pre Few generally performs better than Pre Zero. As shown in Figure 4.1, this improvement is particularly noticeable in the Mobility category, where Pre Few has shown better performance gains. However, while Pre Few captures more refined aspects of the content, the overall structural alignment with the reference content remains similar between the two models.

To summarize, the performance of FT Zero and FT Few outperforms both the non-fine-tuned baseline and the non-fine-tuned models using zero-shot and few-shot prompting methods.

Furthermore, within the fine-tuning system, FT Zero generally demonstrates better performance compared to FT Few. Within prompting system, the few-shot prompting tends to have slightly better performance relative to zero-shot prompting, except for the category of Communication.

4.3 Category-wise Comparison

To perform a category-wise comparison, I calculated the average scores across the two question types, focusing solely on category performance.

Table 4.6 provides the corresponding average scores using BERT scores. Figure 4.3 illustrates the overall trend across the three categories, showing BERT scores as well as average scores for the four models.

Category	Metric	FT Zero	FT Few	Pre Zero	Pre Few
Self-care	Precision	0.8670	0.8664	0.8619	0.8662
	Recall	0.8776	0.8776	0.8707	0.8726
	F1	0.8721	0.8719	0.8662	0.8692
Communication	Precision	0.8749	0.8731	0.8707	0.8644
	Recall	0.8812	0.8814	0.8721	0.8720
	F1	0.8780	0.8771	0.8713	0.8681
Mobility	Precision	0.8694	0.8659	0.8634	0.8672
	Recall	0.8754	0.8736	0.8662	0.8711
	F1	0.8722	0.8696	0.8647	0.8690

Table 4.4: Average BERT Scores for Four Systems Across Categories

Communication consistently demonstrates the highest performance across all systems. FT Zero leads with an impressive F1-score of 0.8780, supported by high precision (0.8749) and recall (0.8812). FT Few follows closely with an F1-score of 0.8771, with precision at 0.8731 and recall at 0.8814. These high scores suggest that fine-tuning, regardless of the prompting method, enhances the model’s ability to generate accurate and relevant responses in Communication. Pre Zero and Pre Few also perform well but exhibit slightly lower scores, with Pre Zero achieving an F1-score of 0.8713, precision of 0.8707, and recall of 0.8721, and Pre Few scoring 0.8681 in F1, with precision of 0.8644 and recall of 0.8720. The high

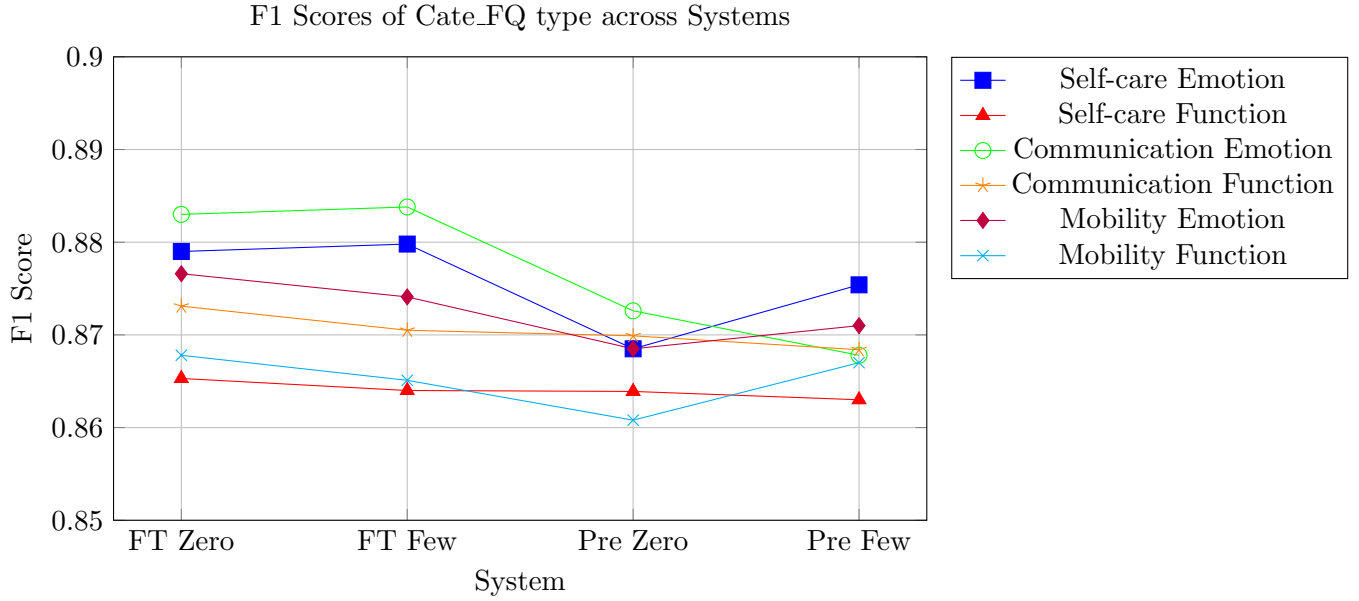


Figure 4.2: F1 Scores of Cate.FQ type across Systems

performance in this category underscores the effectiveness of fine-tuning for generating precise and comprehensive responses.

In Self-care, the performance is slightly lower compared to Communication. FT Zero achieves the highest F1-score of 0.8721, with precision of 0.8670 and recall of 0.8776. FT Few follows closely with an F1-score of 0.8719, precision of 0.8664, and recall of 0.8776. This indicates that while the fine-tuned models perform well, the results are less pronounced compared to Communication. Pre Zero shows an F1-score of 0.8662, with precision of 0.8619 and recall of 0.8707. Pre Few performs slightly better with an F1-score of 0.8690, precision of 0.8662, and recall of 0.8726. These results reflect that while Self-care questions are addressed effectively, the improvements are not as significant as in Communication.

Mobility exhibits the lowest performance across the systems. FT Zero leads with an F1-score of 0.8722, precision of 0.8694, and recall of 0.8754. FT Few has a slightly lower F1-score of 0.8696, with precision of 0.8659 and recall of 0.8736. Pre Zero and Pre Few both show lower scores, with Pre Zero having an F1-score of 0.8647, precision of 0.8634, and recall of 0.8662, and Pre Few scoring 0.8690 in F1, with precision of 0.8672 and recall of 0.8711. The lower scores in Mobility across all models highlight this category as the most challenging, suggesting that further enhancements may be necessary to improve performance in generating relevant responses.

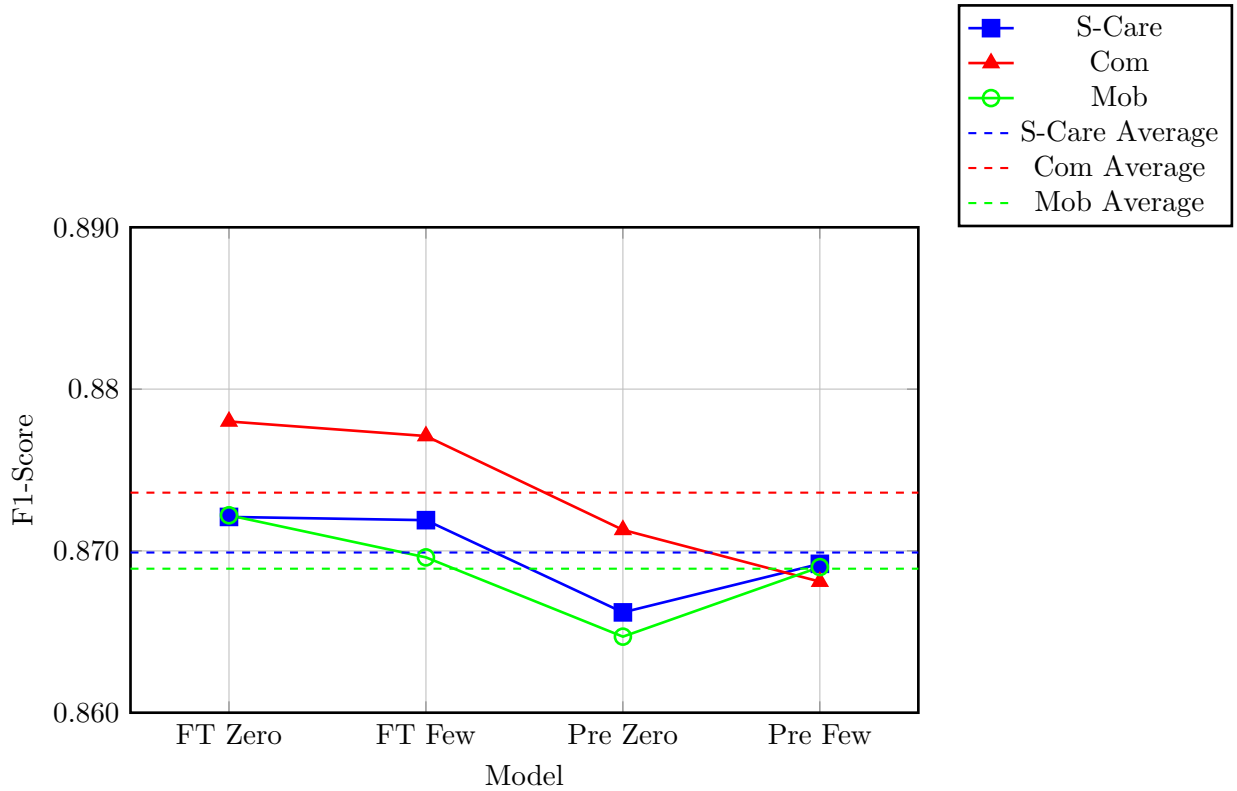


Figure 4.3: F1-Scores for Each Model and Average F1-Scores for Each Category

When calculating the average scores of each category over all models, the results are: Communication: 0.8736, Self-care: 0.8699, Mobility: 0.8689, as indicated by Figure 4.3.

Overall, Communication is the strongest category, achieving the highest precision, recall, and F1-scores across all models. Self-care shows strong performance but with less pronounced results compared to Communication, while Mobility remains the most challenging category, requiring further improvements to match the performance seen in Communication and Self-care.

4.4 FQ type-wise Comparison

This section compares the results based on two types of follow-up questions: function-level questions (FQs) and emotional feedback questions. Table 4.6 presents the results using BERT scores, with a primary focus on BERT-F1 scores for evaluation.

Category	Metric	FT Zero	FT Few	Pre Zero	Pre Few
FQ on Emo	Precision	0.8766	0.8752	0.8677	0.8691
	Recall	0.8827	0.8835	0.8723	0.8739
	F1	0.8795	0.8792	0.8699	0.8714
FQ on Func	Precision	0.8643	0.8629	0.8629	0.8628
	Recall	0.8734	0.8715	0.8670	0.8698
	F1	0.8687	0.8665	0.8655	0.8661

Table 4.5: Average BERT Scores for Four Systems Across Categories

Category	Metric	FT Zero	FT Few	Pre Zero	Pre Few
Self-care	Precision	0.8670	0.8664	0.8619	0.8662
	Recall	0.8776	0.8776	0.8707	0.8726
	F1	0.8721	0.8719	0.8662	0.8692
Communication	Precision	0.8749	0.8731	0.8707	0.8644
	Recall	0.8812	0.8814	0.8721	0.8720
	F1	0.8780	0.8771	0.8713	0.8681
Mobility	Precision	0.8694	0.8659	0.8634	0.8672
	Recall	0.8754	0.8736	0.8662	0.8711
	F1	0.8722	0.8696	0.8647	0.8690

Table 4.6: Average BERT Scores for Four Systems Across Categories

The comparison between question types of emotional feedback and function level measuring reveals a clear trend in model performance. Across all four systems—FT Zero, FT Few, Pre Zero, and Pre Few—Emotion-related questions consistently achieve higher average BERT-F1 scores compared to Function-related questions, as illustrated in Figure 4.4 below.

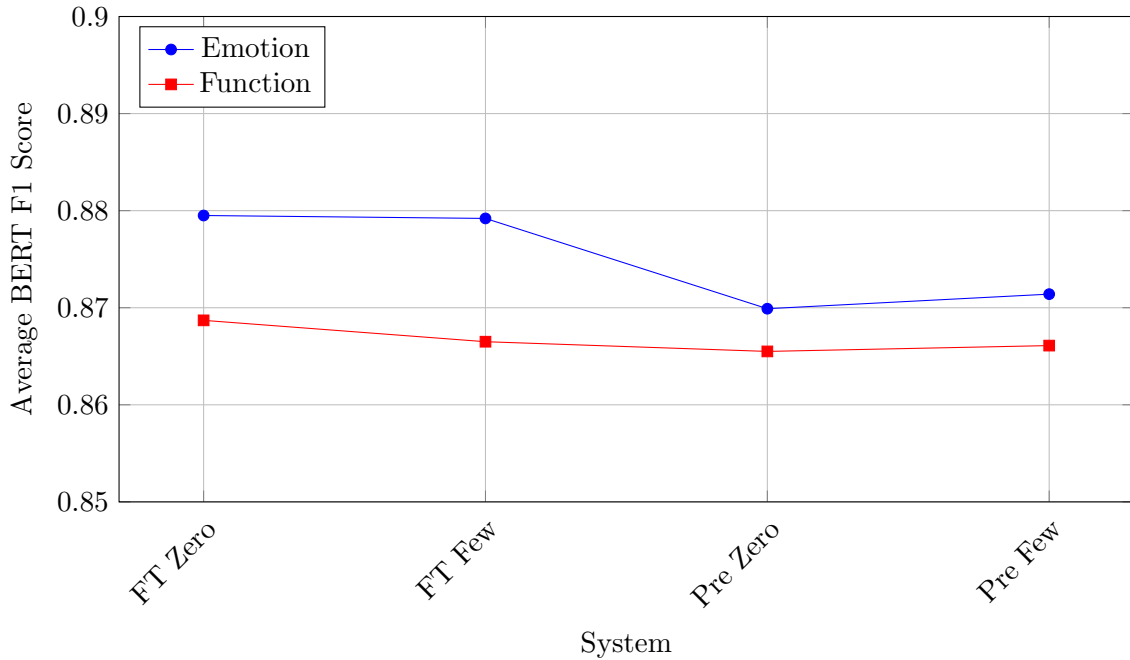


Figure 4.4: Comparison of Average BERT F1 Scores for Emotion and Function FQ Types across Different Systems

For Emotion-related questions, the average F1 scores are notably higher. For example,

the FT Zero system, which performs best in this category, achieves an F1 score of 0.8795. This trend is consistent across all systems, with Emotion-related F1 scores ranging from 0.8714 to 0.8795, reflecting robust performance in this category. In contrast, Function-related questions consistently exhibit lower F1 scores. The highest score achieved by the FT Zero system for Function questions is 0.8687, which is lower than the scores for Emotion-related questions across the same systems. The F1 scores for Function questions are relatively close across the four systems, with FT Few, Pre Zero, and Pre Few showing scores between 0.8655 and 0.8687. Overall, the data highlights a disparity in performance between Emotion and Function question types, with Emotion questions being handled with higher accuracy across all evaluated systems.

Chapter 5

Error Analysis and Discussion

This Chapter performs error analysis and discussion of the finding from results.

5.1 Error Analysis

Error analysis is crucial in natural language processing tasks such as question generation because errors directly impact the effectiveness and reliability of the generated content. When questions are poorly generated, error analysis helps identify and rectify these issues, providing insights and valid assessments.

Continuing a conversation has always been challenging, as it requires identifying key concepts worth exploring and asking relevant questions to extract targeted information Hu et al. (2024); Xiao et al. (2020a). In generating follow-up questions (FQs), the model is expected to identify the target activity in the given conversations and subsequently ask questions aligned with the task objectives of monitoring functional levels and emotional feedback. Moreover, in natural language generation tasks, including question generation, it is essential to evaluate the relevance, completeness and coherence of the generated text. Therefore, the current error analysis mainly involves three parts: errors in identifying target activities, errors in generating task-objective-relevant FQs, and errors related to the quality of generated text in terms of relevance, completeness and coherence.

5.1.1 Errors in Identifying Target Activities

The ability to detect target activities in conversations is crucial for question generation, as it directly impacts the final evaluation, especially if the model’s focus differs from that of human annotators. Errors in identifying the target activity can be grouped into several scenarios.

The first scenario involves identifying a different activity than the one in the reference. While this type of error is not necessarily “wrong”—given that there is only one reference per utterance (as discussed in the Limitation section)—it still counts as an error because it differs from the reference and affects accuracy. This can occur when multiple activities are mentioned in a conversation, all of which are worth exploring. In such cases, models may struggle to choose the most relevant one for follow-up question (FQ) generation. For instance, in a conversation discussing “a neighbor’s cat visit”, “crossword puzzles”, and “a favorite game show”, the FT Zero model (the top performer) chose to continue the conversation with the first two activities, while the human reference focused on the third.

Conversation:

C: How was your day? Did anything interesting happen? P: Oh, yes! *My neighbor’s cat visited me* today. It was very friendly and happened to stay all

evening. I enjoyed its company. C: That’s lovely to hear! Have you been *doing your crossword puzzles like usual*? P: Yes, indeed. I completed one this morning. It helps me with keeping my mind sharp. C: That’s wonderful to hear. Did you *watch your favorite game show* in the evening? P: Oh yes, it’s a favorite of mine, keeps me entertained. I try to guess the answers before they’re revealed. It’s quite fun!

Reference:

1. What is your main challenge in *answering to the game shows questions*?
2. Do you read the newspaper to keep up with economy?
3. If you would like to be able to answer the economy questions you might want to consider reading the economy pages as well.

FT Zero Candidate ¹:

1. That’s great to hear about your *neighbor’s cat visiting*. How did you feel when it stayed all evening?
2. Did the cat’s company help you feel more relaxed or energized?
3. That’s wonderful to hear. How about the *crossword puzzle you completed* earlier today, did it help you feel more engaged or focused?
4. That’s great to hear. Did you find it challenging or did the answers come easily to you?

The second scenario involves identifying zero activities worth exploring. When a model incorrectly assumes there is no activity worth probing, it may continue the conversation without asking about any specific activity. For example, in the conversation below, at least three activities are mentioned that could be further explored: ”changing basic body positions”, ”having some trouble getting up from my armchair” and ”watching a favorite show”. FT Zero and FT Few correctly identify ”moving from the armchair to a standing position” as the target activity, while Pre Zero starts a new conversation with open-ended questions, failing to mention any activity.

Conversation:

C: How was your day, Ms. Elliot? Did you have any difficulties *changing basic body positions* today? P: Yes, dear. I have been *having some trouble getting up from my armchair*. I feel a bit stiff. C: I see. I can assist you with some gentle stretches that may help loosen you up. Should we try those tomorrow? P: That would be wonderful, thank you. I really want to regain some flexibility. C: I’m glad to hear that. Remember, little by little progress is good progress! P: Thank you for the encouragement. I really appreciate your help and support.

Reference:

1. were you able to *get in and out of your wheelchair* today?
2. oh, that is worrying. what happened?
3. are you sure your leg is not broken after falling?

FT Zero candidate:

1. Ms. Elliot, did you find it challenging to *move from the armchair to a standing position* today?

¹the candidate generated by fine-tuned models on zero-shot prompted data

2. How did it feel when you tried to *get up from the armchair*?
3. Did you experience any pain or discomfort when trying to stand up?
4. Would you say you have any difficulty with coordination when changing body positions?

FT Few candidate ²:

1. Did you find it difficult to *get up from the armchair* today, Ms. Elliot?
2. Did you feel any discomfort while getting up from the armchair?
3. How long did it take you to get up from the armchair?
4. Did you need any help or assistance to get up from the armchair?

Pre Zero candidate ³:

1. What did you do today?
2. How was it?
3. That's great!
4. Oh, I see. What happened?
5. That's understandable. It can take some time to get used to a new routine.
6. What do you think we could do to help?

Pre Few candidate ⁴:

1. Did you manage to *get up from the chair* today Mrs Elliot?

The third scenario is identifying an unrelated activity. The task aims to generate questions about activities to elicit responses reflecting emotions or function levels. If the chosen activity has little potential for generating such follow-up questions, it is considered unrelated and thus an error. For example, in an output intended to assess function level, FT Zero's first question asks "the neighbor's cat visiting" and Pre Zero asks "the cat", both focused on *My neighbor's cat visited me* in the conversation (for original conversation, see 5.1.1) as the activity, whereas the reference asks *answering to the game shows questions*. Therefore, these two inferences are unrelated to the task and therefore classified as an error.

Reference:

- What is your main challenge in *answering to the game shows questions*?

FT Zero Candidate ⁵:

- That's great to hear about your *neighbor's cat visiting*. How did you feel when it stayed all evening?

Pre Zero candidate:

- What was *the cat* like?

²the candidate generated by fine-tuned models on few-shot prompting

³the candidate generated by pretrained models on zero-shot prompting

⁴the candidate generated by fine-tuned models on few-shot prompting

⁵the candidate generated by fine-tuned models on zero-shot prompted data

5.1.2 Errors in Generating Task-Objective-Relevant FQs

The ability to generate task-objective-relevant follow-up questions (FQs) means that the generated FQs should effectively fulfill the task’s objective, such as eliciting responses that reflect functional levels or emotional feedback. Errors in this area can be classified into several categories.

The first category involves limited information provided by the activity in the conversation for asking. In some cases, even when a specific activity is clearly identified, the activity may be too general to generate FQs that meet the task objectives. While human experts recognize the need for a more specific activity and continue the conversation accordingly, models often fail to make this distinction and ask questions about the general activity instead. This typically occurs because the model does not fully grasp why the activity is worth exploring further. For instance, in a conversation about “understanding a message without words”, the model might ask directly about this general concept. In contrast, human experts would focus on a more specific, related event, such as “(what can you tell when) seeing the face of the one you are talking to”, which can be considered a sub-activity of “understanding a message without words”.

Conversation:

C: Have you ever needed to *understand a message without words*, just through symbols or gestures? P: Oh yes, definitely. Sometimes I can tell if someone is feeling sad just from their body language. C: It’s impressive how we can comprehend messages without any words, isn’t it? P: Absolutely. It’s a remarkable aspect of communication that often gets overlooked. C: It truly is amazing how much information we can gather non-verbally, like knowing when someone is excited based on their gestures. P: Yes, it really adds depth to our interactions and understanding of others.

Reference:

1. when on the phone, do you *miss seeing the face of the one you are talking to*.
2. I see. You can *read a lot from just looking* at people when talk to them.
3. what do you do when you see that happen?

FT Zero candidate:

1. Can you recall a specific situation where you *understood a message without words*?
2. How did you feel during that conversation? Did it feel different from usual conversations?
3. What do you think makes it easier or harder to understand messages without words?

The second category involves the model asking non-target questions. Models may generate questions that do not align with the task objectives, such as monitoring functional levels or eliciting emotional feedback. For example, in a conversation (5.1.1) where the model is expected to inquire about functional levels based on activities like *doing your crossword puzzles* or *watching your favorite game show*, it might instead ask about feelings, as seen in the FT Few candidate, which asks, *How did you feel about...* and *Do you enjoy...* This diverges from the intended task objective. Similarly, models might fail to detect the need for emotional feedback in FQs, resulting in irrelevant or superficial questions that do not

contribute to the task. For instance, in a task designed to elicit emotional feedback, the model might instead focus on functional levels, leading to questions that are off-target.

FT Few candidate:

1. *How did you feel about having company with your neighbor's cat?*
2. Did you enjoy looking after it and making sure it was comfortable?

Similarly, models might fail to detect the emotional feedback required for FQs, leading to irrelevant or superficial questions that do not contribute to the task objective. The following two inference is extracted from the output of task to ask emotional feedback whereas these two asks about the function level.

Pre Zero candidate:

1. What was the most difficult part of the lunges for you today?
2. Did you have any difficulties with the other exercises as well?
3. How did you manage to complete the squats?

Pre Few candidate:

1. That's great to hear! Did you find it easy to follow the therapist's instructions during the session?
2. It's completely okay to ask for clarification when needed. Is there any particular exercise that you found most challenging to comprehend today?

5.1.3 Relevant, Complete and Coherent

Relevance is a critical aspect of natural language generation, including question generation. In the context of generating follow-up questions (FQs) between caregivers and patients for monitoring purposes, being complete and coherent is equally important. This is because such conversations, in real-world scenarios, are typically complete and coherent—a series of questions that build on one another, differing from more spontaneous natural conversations. Therefore, ensuring that FQs are relevant, complete, and coherent is crucial, yet it remains a challenge for generative models, often leading to errors.

For instance, in the following conversation, the Pre Zero model generates repetitive questions without fully exploring the context of the conversation. This lack of completeness and coherence can adversely affect the model's accuracy.

Conversation:

C: How are you feeling today? Did you have any troubles going to the bathroom?

P: I'm feeling fine thank you. No issues with using the bathroom today.

Pre Zero candidate:

1. What are you feeling today? Did you have any troubles going to the bathroom?
2. What are you feeling today? Did you have any troubles going to the bathroom?
3. What are you feeling today? Did you have any troubles going to the bathroom?
4. are you feeling today? Did you have any troubles going to the bathroom?]

Pre Few candidate:

- How did you go to the bathroom today?

5.2 Discussion of Results

5.2.1 Limitation

Several limitations affected the outcomes of this task.

Firstly, the absence of a validation dataset during prompt engineering had a significant impact on the entire downstream process. Prompt engineering is critical in this context because it directly influences the generation of training data, which is essential for fine-tuning models. The lack of a validation dataset likely hindered the optimization of model performance, as it restricted the ability to fine-tune prompts and evaluate their effectiveness. Although human evaluation was conducted, relying on a single annotator reduces the reliability of the results compared to an automatic evaluation with a validation dataset. Additionally, the prompts developed during prompt engineering were consistently used across both fine-tuned systems and default systems for final inference generation. While this consistency helps maintain uniformity, the absence of validation data limited opportunities for further refining model performance. Validation data would have been particularly useful in enhancing few-shot prompting, where the selection and order of examples play a crucial role. This limitation is reflected in the performance of fine-tuned models using few-shot prompting training data and in pretrained models using few-shot prompting, where the limited guidance from only two examples may have constrained the diversity and effectiveness of the training process.

Secondly, there was a shortage of references for each inference. As outlined in the methodology and discussed in the error analysis, multiple references were intended to improve evaluation accuracy by accommodating the one-to-many nature of conversations, which can lead to various valid inference possibilities. However, due to time constraints, only one reference was provided per inference, potentially compromising the accuracy of the evaluation. This shortage also exacerbated the risk of fake errors—incorrect error annotations that misidentify a model’s output as wrong when it is, in fact, a valid alternative—because the evaluation lacked the necessary context to properly assess all possible valid responses.

Thirdly, the training data lacked diversity. Despite the broad range of daily activities defined by the International Classification of Functioning, Disability, and Health (ICF), only three categories were selected for this task due to time constraints. This limited scope could impair the models’ ability to generalize to conversations about topics not covered during training, potentially affecting overall performance.

5.2.2 System-wise

The primary system comparison is between the fine-tuned systems and Non-fine-tuned systems. Fine-tuned systems include fine-tuning llama3 on datasets generated by GPT using zero-shot prompting and few-shot prompting. After fine-tuning, llama models were tested using basic instructions, without employing additional few-shot or zero-shot prompts during the inference. The Non-fine-tuned systems include the pre-trained llama3 with zero-shot prompting and few-shot prompting during the inference.

The Fine-tuned models on few-shot and zero-shot prompting performed similarly, with the model fine-tuned on zero-shot data even slightly outperforming the one fine-tuned on few-shot data. This may be because the zero-shot-generated training data was more diverse and of higher quality. During prompt engineering, only two examples were used for few-shot prompts, and these examples remained the same across the individual category. This might limit the fine-tuned models’ ability to handle activities outside the provided activities. For instance, under the category of ”mobility”, follow-up questions (FQs) were generated around activities like ”Walking”, ”Going up and downstairs”, and ”Moving around”. However, the two FQ examples used in few-shot prompting were exclusively about ”walking”, potentially

limiting the diversity of the few-shot training data by providing task guidance with limited examples and further influencing the performance of fine-tuned llama trained on few-shot prompted data.

This suggests that fine-tuning on diverse data, whether generated by few-shot or zero-shot prompts, allows the model to effectively learn the underlying task. The slight performance edge observed with zero-shot data might indicate that it offered more generalized or varied examples, enhancing the model’s ability to generalize across different scenarios and it stressed the importance of example choice during few-shot prompting.

In contrast, the second system group includes non-fine-tuned LLaMA3 models evaluated using few-shot and zero-shot prompting. In this case, the models performed better with few-shot prompts compared to zero-shot prompts. This outcome is expected, as few-shot prompting provides explicit examples of the task, offering additional context and reducing ambiguity. This guided approach helps the model better understand and execute the task. On the other hand, zero-shot prompting relies entirely on the model’s pre-existing knowledge without specific examples, which can lead to less accurate performance, particularly if the task or context is not well-defined.

Overall, the finding that fine-tuned LLaMA3 models generally outperform non-fine-tuned models underscores the benefits of task-specific adaptation. Fine-tuning allows models to internalize and specialize in the task, leading to better generalization and more accurate results, even when evaluated with minimal instructions. In contrast, while prompting can improve the performance of non-fine-tuned models—as evidenced by comparisons with the baseline model, an original non-fine-tuned LLaMA3 evaluated without any prompting—it cannot match the deep task alignment achieved through fine-tuning, which is essential for optimal performance.

5.2.3 Category-wise

The results indicate the finding of category-related trend that models generally have higher accuracy in the communication category, followed by self-care, and then mobility.

Despite having the most training data in the mobility category (375 examples) and the highest number of activity types (16), this category does not exhibit the highest accuracy. This might be due to the complexity and variability inherent in having so many types of activities. The model may struggle with this diversity, as the vast range of examples could introduce more noise and make it harder for the model to generalize effectively. Consequently, the accuracy is lower despite the extensive training data.

In contrast, the communication category, which has fewer training examples (222) and fewer activity types (13), demonstrates higher accuracy. When comparing the ICF activities used to generate conversations and FQs, the activities within the communication category, while not the least in amount, exhibit less variation. For instance, to assess functional levels related to mobility, the training data conversations might involve distinct and diverse activities such as "hand and arm use" and "moving around in different locations". In contrast, activities related to communication are generally variations of similar tasks, such as "receiving spoken messages" and "producing non-verbal languages". This may suggest that the tasks within the communication category might be more uniform or straightforward for the model to handle, allowing it to achieve better performance even with less training data. Another possibility is that this type of conversations and questions is more likely to be part of the pretraining data and thus the model is already better equipped to represent and generate this category. Given that the test data maintains the same distribution in terms of data quantity and types, it’s likely that the reduced complexity and fewer activity types in the communication category make it easier for the model to learn and generalize from the available examples.

The self-care category, with the least amount of training data (166 examples) and the fewest activity types (7), ranks in between in terms of accuracy. Similar to the communication category, the fewer activity types might present a more straightforward task for the model compared to the mobility category, although the limited data could constrain the model’s ability to learn comprehensively.

5.2.4 FQ type-wise

The finding that function-level follow-up questions (FQs) have lower performance compared to emotional feedback FQs across all categories highlights the varying complexity and demands of these two types of questions. Function-level FQs, designed to probe specific functional aspects of a patient’s condition, require a deep understanding of medical knowledge to ask appropriate questions. The complexity of these questions lies in accurately assessing and interpreting the patient’s functional abilities and limitations. This type of questioning demands precision and a thorough understanding of the patient’s medical context, making it more challenging for models to generate or interpret these questions accurately. The lower performance in this area likely reflects the model’s difficulty in handling the nuanced and specialized language required for effective function-level questions.

In contrast, FQs related to emotional feedback are generally less complex and more straightforward. These questions aim to gather information about the patient’s feelings, perceptions, or emotional responses to the activities discussed in the conversation. Emotional feedback questions are more about understanding the patient’s perspective rather than assessing specific functional abilities, requiring less technical precision and medical knowledge. The relative simplicity of these questions allows the model to generate and interpret them more effectively, leading to higher performance in this area. This discrepancy in performance underscores the challenges models face when dealing with more specialized, knowledge-intensive tasks, compared to simpler, more general inquiries.

Chapter 6

Conclusion and Future Work

This project aimed to develop an automatic method for generating appropriate follow-up questions to monitor patients' physical and mental conditions using the Llama3 generative model. The study explored two optimization approaches: prompt engineering (utilizing zero-shot and few-shot prompting) and fine-tuning the model. The primary focus was on comparing two systems: fine-tuned (**FT**) models that did not use prompt engineering and non-fine-tuned (**non-FT**) models that relied on prompt engineering techniques. Due to constraints such as the substantial amount of training data required for fine-tuning and limited time, GPT-3.5 was used to generate training data through zero-shot and few-shot prompting.

There are four findings in the current project:

1. **Effectiveness of Fine-Tuning:** The results demonstrated that fine-tuned LLaMA3 models generally outperformed non-fine-tuned models, underscoring the benefits of task-specific adaptation. Fine-tuning allowed the models to internalize and specialize in the task, leading to better generalization and more accurate results, even with minimal instructions.
2. **Impact of Prompting:** While prompting improved the performance of non-fine-tuned models, as evidenced by comparisons with the baseline model, it did not match the deep task alignment achieved through fine-tuning.
3. **Category-Related Performance Trends:** The models generally showed higher accuracy in the communication category, followed by self-care and then mobility. This trend suggests that the quantity, quality, and distribution of training data significantly influence performance, with less diversity in training data proving more effective in certain closed-domain scenarios.
4. **Difficulty in Generating Function-Level Follow-Up Questions:** The study found that function-level follow-up questions (FQs) had lower performance compared to emotional feedback FQs across all categories. This disparity highlights the varying complexity and demands of these two types of questions, with function-level FQs requiring a deeper understanding of medical knowledge.

Despite those findings, the project faced several limitations: the absence of a validation dataset during prompt engineering may make it challenging to refine prompts effectively, potentially leading to sub-optimal performance; a shortage of references for each inference may bias the performance assessment, as it did not fully capture the range of acceptable responses; and a lack of diversity in training data may affect the model's ability to generalize across different conversational topics and patient interactions.

Addressing these limitations can be beneficial for future work. Future research should incorporate validation datasets, expand reference data and enhance training data diversity, which will advance the development of reliable models for generating follow-up questions in patient monitoring, ultimately contributing to better healthcare outcomes. Additionally, models were trained on data generated by GPT, and the resulting performance of Llama3 suggests the feasibility of using LLMs to generate data for subsequent fine-tuning. This approach is promising for future work, as it can mitigate the challenge of obtaining extensive training data, which is often resource-intensive.

What's more, the primary results indicate that fine-tuned systems generally achieve higher scores than non-fine-tuned systems across the three categories of communication, self-care, and mobility, as well as for the two follow-up question types for measuring function levels and requiring emotional feedback. This trend holds true whether the systems are evaluated using traditional metrics or BERT scores, highlighting the effectiveness of fine-tuning in generating follow-up questions.

During error analysis, it was observed that fine-tuned models, whether trained on zero-shot or few-shot prompted data, excel in identifying activities worth probing and in generating relevant, complete, and coherent follow-up questions that align with the task objectives, such as health monitoring. Thus, applying generative models with fine-tuning approaches shows promise for future use in healthcare, particularly for monitoring patients' physical and mental conditions.

Bibliography

- S. Arora, A. Narayan, M. F. Chen, L. Orr, N. Guha, K. Bhatia, I. Chami, F. Sala, and C. Ré. Ask me anything: A simple strategy for prompting language models, 2022. URL <https://arxiv.org/abs/2210.02441>.
- L. Athota, V. K. Shukla, N. Pandey, and A. Rana. Chatbot for healthcare system using artificial intelligence. In *2020 8th International conference on reliability, infocom technologies and optimization (trends and future directions)(ICRITO)*, pages 619–622. IEEE, 2020.
- Y. Bengio. Practical recommendations for gradient-based training of deep architectures, 2012a. URL <https://arxiv.org/abs/1206.5533>.
- Y. Bengio. Deep learning of representations for unsupervised and transfer learning. In I. Guyon, G. Dror, V. Lemaire, G. Taylor, and D. Silver, editors, *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*, pages 17–36, Bellevue, Washington, USA, 02 Jul 2012b. PMLR. URL <https://proceedings.mlr.press/v27/bengio12a.html>.
- T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners, 2020. URL <https://arxiv.org/abs/2005.14165>.
- Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P. S. Yu, Q. Yang, and X. Xie. A survey on evaluation of large language models, 2023.
- B. Chen, Z. Zhang, N. Langrené, and S. Zhu. Unleashing the potential of prompt engineering in large language models: a comprehensive review, 2024. URL <https://arxiv.org/abs/2310.14735>.
- F. Clarizia, F. Colace, M. Lombardi, F. Pascale, and D. Santaniello. Chatbot: An education support system for student. In *Cyberspace Safety and Security: 10th International Symposium, CSS 2018, Amalfi, Italy, October 29–31, 2018, Proceedings 10*, pages 291–302. Springer, 2018.
- S. Cozzi, A. Martinuzzi, and V. Della Mea. Ontological modeling of the international classification of functioning, disabilities and health (icf): Activitiesparticipation and environmental factors components. *BMC Medical Informatics and Decision Making*, 21, 12 2021. doi: 10.1186/s12911-021-01729-x.
- M. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, C. Monz, M. Post, and L. Specia, editors, *Proceedings of the Ninth Workshop on Statistical*

- Machine Translation*, pages 376–380, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-3348. URL <https://aclanthology.org/W14-3348>.
- J. Deriu, A. Rodrigo, A. Otegi, G. Echegoyen, S. Rosset, E. Agirre, and M. Cieliebak. Survey on evaluation methods for dialogue systems. *Artificial Intelligence Review*, 54(1): 755–810, June 2020. ISSN 1573-7462. doi: 10.1007/s10462-020-09866-x. URL <http://dx.doi.org/10.1007/s10462-020-09866-x>.
- J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL <https://aclanthology.org/N19-1423>.
- P. F. Edemekong, D. Bomgaars, S. Sukumaran, and S. B. Levy. Activities of daily living, 2019. StatPearls. Retrieved from https://digitalcollections.dordt.edu/faculty_work/1222.
- I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- M. Heilman and N. A. Smith. Good question! statistical ranking for question generation. In R. Kaplan, J. Burstein, M. Harper, and G. Penn, editors, *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617, Los Angeles, California, June 2010. Association for Computational Linguistics. URL <https://aclanthology.org/N10-1086>.
- G. Hiremath, A. Hajare, P. Bhosale, R. Nanaware, and K. Wagh. Chatbot for education system. *International Journal of Advance Research, Ideas and Innovations in Technology*, 4(3):37–43, 2018.
- S. Höglund and J. Khedri. Comparison between rlhf and rlai in fine-tuning a large language model, 2023.
- O. Honovich, T. Scialom, O. Levy, and T. Schick. Unnatural instructions: Tuning language models with (almost) no human labor, 2022. URL <https://arxiv.org/abs/2212.09689>.
- J. Howard and S. Ruder. Universal language model fine-tuning for text classification. In I. Gurevych and Y. Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1031. URL <https://aclanthology.org/P18-1031>.
- E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models, 2021. URL <https://arxiv.org/abs/2106.09685>.
- J. Hu, J. Guo, N. Tang, X. Ma, Y. Yao, C. Yang, and Y. Xu. Designing the conversational agent: Asking follow-up questions for information elicitation. *Proc. ACM Hum.-Comput. Interact.*, 8(CSCW1), apr 2024. doi: 10.1145/3637320. URL <https://doi.org/10.1145/3637320>.

- J. Kaddour, J. Harris, M. Mozes, H. Bradley, R. Raileanu, and R. McHardy. Challenges and applications of large language models, 2023. URL <https://arxiv.org/abs/2307.10169>.
- L. Kelvin. Icf-based classification and measurement of functioning. *European journal of physical and rehabilitation medicine*, 44:315–28, 2008.
- Y.-C. Lee, N. Yamashita, Y. Huang, and W. Fu. "i hear you, i feel you": Encouraging deep self-disclosure through a chatbot. pages 1–12, 04 2020. doi: 10.1145/3313831.3376175.
- C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL <https://aclanthology.org/W04-1013>.
- C.-W. Liu, R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation, 2017. URL <https://arxiv.org/abs/1603.08023>.
- P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, 2021. URL <https://arxiv.org/abs/2107.13586>.
- Y. Lu, M. Bartolo, A. Moore, S. Riedel, and P. Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.556. URL <https://aclanthology.org/2022.acl-long.556>.
- T. Makasi, A. Nili, K. Desouza, and M. Tate. Chatbot-mediated public service delivery: A public service value-based framework. *First Monday*, 25(12), 2020.
- N. Motalebi, E. Cho, S. S. Sundar, and S. Abdullah. Can alexa be your therapist? how back-channeling transforms smart-speakers to be active listeners. In *Companion Publication of the 2019 Conference on Computer Supported Cooperative Work and Social Computing*, page 309–313, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450366922. doi: 10.1145/3311957.3359502. URL <https://doi.org/10.1145/3311957.3359502>.
- OpenAI. Moving from completions to chat completions in the openai api. <https://help.openai.com/en/articles/7042661-moving-from-completions-to-chat-completions-in-the-openai-api>. Accessed: August 11, 2024.
- OpenAI. Gpt-3.5 turbo, 2024. URL <https://platform.openai.com/docs/models/gpt-3-5-turbo>. Accessed: 2024-08-26.
- OpenAI. Gpt-4 and gpt-4 turbo models, 2024. URL <https://platform.openai.com/docs/models/gpt-4-turbo-and-gpt-4>. Accessed: 2024-08-26.
- OpenAI. Chatgpt, 2024. URL <https://openai.com/chatgpt>. Accessed: 2024-08-26.
- W. H. Organization. *International Classification of Functioning, Disability and Health: ICF*. World Health Organization, 2001. URL <https://iris.who.int/handle/10665/42407>. Accessed: 2024-08-10.

- K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL <https://doi.org/10.3115/1073083.1073135>.
- A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019. URL <https://api.semanticscholar.org/CorpusID:160025533>.
- V. Rus, B. Wyse, P. Piwek, M. Lintean, S. Stoyanchev, and C. Moldovan. The first question generation shared task evaluation challenge. In J. Kelleher, B. M. Namee, and I. v. d. Sluis, editors, *Proceedings of the 6th International Natural Language Generation Conference*. Association for Computational Linguistics, July 2010. URL <https://aclanthology.org/W10-4234>.
- V. Rus, B. Wyse, P. Piwek, M. Lintean, S. Stoyanchev, and C. Moldovan. Question generation shared task and evaluation challenge – status report. In C. Gardent and K. Striegnitz, editors, *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 318–320, Nancy, France, sep 2011. Association for Computational Linguistics. URL <https://aclanthology.org/W11-2853>.
- M. Shanahan, K. McDonell, and L. Reynolds. Role-play with large language models, 2023. URL <https://arxiv.org/abs/2305.16367>.
- H.-Y. Shum, X. He, and D. Li. From eliza to xiaoice: Challenges and opportunities with social chatbots. *Frontiers of Information Technology and Electronic Engineering*, 19, 01 2018. doi: 10.1631/FITEE.1700826.
- A. Stamatis, A. Gerontas, A. Dasyras, and E. Tambouris. Using chatbots and life events to provide public service information. In *Proceedings of the 13th International Conference on Theory and Practice of Electronic Governance*, pages 54–61, 2020.
- G. Stucki, N. Kostanjsek, B. Ustün, and A. Cieza. Icf-based classification and measurement of functioning. *European journal of physical and rehabilitation medicine*, 44(3):315–328, 2008.
- M.-H. Su, C.-H. Wu, K.-Y. Huang, and C.-K. Chen. Attention-based dialog state tracking for conversational interview coaching. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6144–6148, 2018. doi: 10.1109/ICASSP.2018.8461494.
- H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models, 2023. URL <https://arxiv.org/abs/2302.13971>.
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023. URL <https://arxiv.org/abs/1706.03762>.
- P. Vossen, S. B. Santamaria, and T. Baier. A conversational agent for structured diary construction enabling monitoring of functioning & well-being, 2024. Working paper.
- J. Wang, E. Shi, S. Yu, Z. Wu, C. Ma, H. Dai, Q. Yang, Y. Kang, J. Wu, H. Hu, C. Yue, H. Zhang, Y. Liu, Y. Pan, Z. Liu, L. Sun, X. Li, B. Ge, X. Jiang, D. Zhu, Y. Yuan, D. Shen, T. Liu, and S. Zhang. Prompt engineering for healthcare: Methodologies and applications, 2024. URL <https://arxiv.org/abs/2304.14670>.

- S. Wang, Y. Liu, Y. Xu, C. Zhu, and M. Zeng. Want to reduce labeling cost? GPT-3 can help. In M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4195–4205, Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.354. URL <https://aclanthology.org/2021.findings-emnlp.354>.
- J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL <https://arxiv.org/abs/2201.11903>.
- Z. Xiao, M. X. Zhou, W. Chen, H. Yang, and C. Chi. If i hear you correctly: Building and evaluating interview chatbots with active listening skills. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–14, New York, NY, USA, 2020a. Association for Computing Machinery. ISBN 9781450367080. doi: 10.1145/3313831.3376131. URL <https://doi.org/10.1145/3313831.3376131>.
- Z. Xiao, M. X. Zhou, Q. V. Liao, G. Mark, C. Chi, W. Chen, and H. Yang. Tell me about yourself: Using an ai-powered chatbot to conduct conversational surveys with open-ended questions. *ACM Trans. Comput.-Hum. Interact.*, 27(3), jun 2020b. ISSN 1073-0516. doi: 10.1145/3381804. URL <https://doi.org/10.1145/3381804>.
- B. Xu, A. Yang, J. Lin, Q. Wang, C. Zhou, Y. Zhang, and Z. Mao. Expertprompting: Instructing large language models to be distinguished experts, 2023. URL <https://arxiv.org/abs/2305.14688>.
- L. Xu, L. Sanders, K. Li, J. C. Chow, et al. Chatbot for health care and oncology applications using artificial intelligence and machine learning: systematic review. *JMIR cancer*, 7(4): e27850, 2021.
- J. Ye, J. Gao, Q. Li, H. Xu, J. Feng, Z. Wu, T. Yu, and L. Kong. Zerogen: Efficient zero-shot learning via dataset generation, 2022. URL <https://arxiv.org/abs/2202.07922>.
- J. Ye, X. Chen, N. Xu, C. Zu, Z. Shao, S. Liu, Y. Cui, Z. Zhou, C. Gong, Y. Shen, J. Zhou, S. Chen, T. Gui, Q. Zhang, and X. Huang. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models, 2023. URL <https://arxiv.org/abs/2303.10420>.
- Y.-T. Yeh, M. Eskenazi, and S. Mehri. A comprehensive assessment of dialog evaluation metrics, 2021. URL <https://arxiv.org/abs/2106.03706>.
- R. Zhang, J. Guo, L. Chen, Y. Fan, and X. Cheng. A review on question generation from natural language text. *ACM Trans. Inf. Syst.*, 40(1), sep 2021. ISSN 1046-8188. doi: 10.1145/3468889. URL <https://doi.org/10.1145/3468889>.
- T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. Bertscore: Evaluating text generation with bert, 2020. URL <https://arxiv.org/abs/1904.09675>.
- Z. Zhang, J. Gao, R. S. Dhaliwal, and T. J.-J. Li. Visar: A human-ai argumentative writing assistant with visual programming and rapid draft prototyping. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23. ACM, Oct. 2023. doi: 10.1145/3586183.3606800. URL <http://dx.doi.org/10.1145/3586183.3606800>.
- T. Zhao, R. Zhao, and M. Eskenazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In R. Barzilay and M.-Y. Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664, Vancouver, Canada, July 2017. Association for

Computational Linguistics. doi: 10.18653/v1/P17-1061. URL <https://aclanthology.org/P17-1061>.

- Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh. Calibrate before use: Improving few-shot performance of language models. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR, 18–24 Jul 2021. URL <https://proceedings.mlr.press/v139/zhao21c.html>.