

**COMBINING PERSONA AND ARGUMENT IN
DIALOGUE**

**A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
MASTER OF SCIENCES
in Computer Engineering**

**by
Şükrü GÜZEL**

**June 2024
İZMİR**

We approve the thesis of **Şükrü GÜZEL**

Examining Committee Members:

Assoc. Prof. Dr. Selma TEKIR

Department of Computer Engineering, İzmir Institute of Technology

Asst. Prof. Dr. Damla OĞUZ

Department of Computer Engineering, İzmir Institute of Technology

Assoc. Prof. Dr. Özgü CAN

Department of Computer Engineering, Ege University

Assoc. Prof. Dr. Selma TEKIR

Department of Computer Engineering
İzmir Institute of Technology

Professor Dr. Cüneyt F. BAZLAMAÇCI

Head of the Department of
Computer Engineering

Professor Dr. Mehtap EANES

Dean of the Graduate School of
Engineering and Sciences

ACKNOWLEDGMENTS

I am grateful to my academic advisor, Associate Professor Dr. Selma Tekir, for her guidance, support, and motivation during this process. I would also like to express my thanks to my wife, Sena Güzel, for her unwavering support, as well as to my family and everyone else who has supported me. Throughout my journey, the support and motivation they provided were a great source of energy for me. The knowledge and experience gained during this time have greatly enriched my life and will forever shape my future.

ABSTRACT

COMBINING PERSONA AND ARGUMENT IN DIALOGUE

The increasing popularity of personalized dialogue systems has gained momentum as people's desire for human-like interaction grows. This thesis aims to increase persona-consistent responses in personalized dialogue systems. A data augmentation method was used to enhance the persona consistency of dialogue systems. This technique utilized Large Language Models' few-shot learning capabilities to add counterfactual sentences to the dialogue. GPT 3.5 and Llama 2 models were used to generate counterfactual sentences using the few-shot prompting method. The augmentation method was applied to every dialogue in the PersonaChat dataset that did not originally contain a counterfactual sentence. Evaluation using the state-of-the-art personalized dialogue generation study showed that the persona-consistency results of the dataset augmented with the GPT 3.5 model showed better performance when assessed using metrics.

ÖZET

KARŞILIKLI KONUŞMADA KARAKTERİN VE GÖRÜŞÜN BİRLEŞTİRİLMESİ

İnsanların, insan benzeri konuşmalara ilgisinin artmasından dolayı son zamanlarda kişiselleştirilmiş diyalog sistemlerine olan ilgi artmaktadır. Bu sistemlerde kişilik bilgilerinin diyalog sistemlerine entegre olmasıyla beraber daha kişisel cevaplar sunulmaktadır. Bu tezde, kişiselleştirilmiş diyalog sistemlerindeki cevapların kişilik tutarlılığının artması amaçlanmıştır ve bu amaca yönelik veri büyütme yöntemi uygulanmıştır. İlgili yöntem ile karşı olgusal cümle içermeyen her diyaloga karşı olgusal cümleler eklenmiş ve bu süreçte Büyük Dil Modellerinin (LLMs) az vuruşlu öğrenme yeteneklerinden yararlanılmıştır. Az vuruşlu öğrenme yöntemi ile karşı olgusal cümleler üretmek için GPT 3.5 ve Llama 2 modelleri seçilmiş olup, PersonaChat veri setinde karşı olgusal cümle içermeyen diyalogların her biri için veri artırma yöntemi uygulanmıştır. Metrikler ile yapılan değerlendirme sonucunda GPT 3.5 modeliyle zenginleştirilen veri setinin kişilik-tutarlılık sonuçlarının daha iyi olduğu gözlemlenmiştir.

TABLE OF CONTENTS

LIST OF FIGURES	Viii
LIST OF TABLES	Ix
CHAPTER 1. INTRODUCTION	1
CHAPTER 2. BACKGROUND	5
2.1. Language Models.....	5
2.1.1 Large Language Models (LLMs).....	10
2.1.2 Pre-training and Fine-tuning	11
2.1.3 Prompting	12
2.1.3.1 Zero-shot Prompting	13
2.1.3.2 Few-shot Prompting	14
2.2. Applications of Language Models.....	15
2.2.1 Natural Language Understanding (NLU)	15
2.2.2 Natural Language Generation (NLG)	16
2.2.3 Natural Language Inference (NLI)	16
2.2.4 Dialogue Systems	17
2.2.4.1 Personalized Dialogue Systems.....	18
2.3. Counterfactual Statements	19
CHAPTER 3. RELATED WORK.....	21
CHAPTER 4. DATASET	25
CHAPTER 5. METHODOLOGY	28

5.1. Counterfactual Sentences Detection in Dataset	28
5.2. Data Preparation	29
5.3. Selection of Few-shot Prompting Examples	30
5.4. Selection of a Large Language Model	34
5.5. Prompt Design	35
5.6. Generating Counterfactual Sentences and Creating Dialogues	36
5.7. Evaluation	36
CHAPTER 6. EXPERIMENTS	39
6.1. Implementation Details	39
6.2. Results and Discussion	41
CHAPTER 7. CONCLUSION	43
7.1. Future Work	44
REFERENCES	44
APPENDIX	56
A.1. Case Study	56

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
Figure 1.1	Dialogue with Counterfactual Sentence Added	3
Figure 2.1	The Transformer - model architecture (Vaswani et al. 2017)	6
Figure 2.2	Scaled Dot-Product Attention on the left. Multi-Head Attention on the right (Vaswani et al. 2017).....	8
Figure 2.3	A Diagram of Text-to-Text Framework (Raffel et al. 2019)	10
Figure 2.4	Real-World Applications of LLMs (Rachel 2022)	11
Figure 2.5	Prompt Example (OpenAI 2024)	12
Figure 2.6	Zero-shot Prompting Example (OpenAI 2024)	14
Figure 4.1	Creation Stages of PersonaChat Dataset (Princeton University 2020)	25
Figure 5.1	Project Workflow	29
Figure 5.2	Evaluation Steps	30
Figure 5.3	Some Counterfactual Sentences Detected From the PersonaChat..	31
Figure 5.4	One of the Few-Shot Prompting Examples	33
Figure 5.5	Fine-tuned Pre-trained NLI Model	37

LIST OF TABLES

<u>Table</u>		<u>Page</u>
Table 2.1	GPT Language Models Comparison	9
Table 2.2	Examples of the Entailment, Neutral and Contradiction.....	17
Table 3.1	Related Work Comparison	24
Table 4.1	Some Dialogue Datasets Statistics	26
Table 5.1	An Example Line From the PersonaChat after Preprocessing.....	32
Table 6.1	Statistics of Generated Counterfactual Sentences	41
Table 6.2	Automatic Evaluation Results Comparison for Persona-based Dia- logue Generation.....	41
Table 6.3	C-score results calculated using pre-trained and pre-trained fine- tuned NLI models.....	42
Table A.1	Example of Response Generation on the PersonaChat.....	56
Table A.2	Example of Response Generation on the PersonaChat.....	57

CHAPTER 1

INTRODUCTION

Enriching and personalizing the user experience in dialogue systems aims to increase the correlation between humans and AI. This can be achieved by incorporating personality traits into dialogue systems. There is a higher interest in the responses created due to increased fluency and human reactions with the persona integrated into the dialogue systems. To increase this interest, the answers given must reflect the persona or not be contrary to the persona. Many studies are ongoing to provide answers to this issue. The explicit integration of personas into the dialogue systems came to the forefront of this study. Adding persona to dialogue agents to create a reliable conversation system will accelerate the integration process into our daily lives.

Datasets containing personal information play a vital role in integrating human conversational styles into dialogue systems. The PersonaChat dataset (Zhang et al. 2018) was created for this purpose to improve personal and natural conversations. Welleck et al. (Welleck et al. 2019) have shown that each response in PersonaChat is not persona-related. In recent years, there have been many studies on open-domain dialogue systems to produce persona-related responses and to produce the response depending on the context of the dialogue. Nevertheless, there are still areas for improvement (Shea and Yu 2023).

One approach to augment increasing consistency is to expand persona sets that contain personality traits to increase consistency (Cao et al. 2022) (M. Kim et al. 2022) (D. Kim et al. 2023) (Ribeiro, Carvalho, and Coheur 2023). Another approach is to augment the dialogue and personal information in the PersonaChat, that is, to expand the dataset. Datasets created with crowd-sourced are costly; expanding the dataset given in the above approach requires consideration in terms of cost. For instance, while we examine the PersonaChat, the data of two crowd-sourced persons, a role-play created with persona

information, and two people who are made to talk in a mutual conversation with the given instructions are examined (Zhang et al. 2018).

In this work, the consistency problem in persona-based dialogue generation is addressed. The core mandate of our research is to examine the effects of consistency on personalized dialogue agents using the data augmentation method by taking advantage of the capabilities of Large Language Models (LLMs).

LLMs are capable of performing complicated tasks, even with a few examples. This capability is made possible by training language models on massive datasets and then customizing them with a minimum number of examples (Brown et al. 2020). To illustrate, advanced LLMs such as GPT-3 can demonstrate high performance in language comprehension, text production, translation, and various language-based tasks with few examples (Radford et al. 2019).

Few-shot learning is considered an effective method for adapting the overall capabilities of language models to specific tasks (Lin et al. 2022). This study aims to get the desired answer with few-shot learning instead of fine-tuning large language models for the cooperative sentence generation task. In this study, counterfactual sentence generation is carried out using the few-shot learning capability of LLMs. Counterfactual sentences describe what would happen in scenarios that did not occur or in different situations. The dataset used in the study was augmented with the counterfactual sentences produced. It is crucial that the integrated counterfactual sentences are related to persona and dialogue history.

In this work, the persona-consistency problem is comprehensively examined. The consistency of personas in personalized dialogue agents can still be improved. As Welleck (Welleck et al. 2019) pointed out, there are still inconsistencies in the answers produced. We are suggesting improvements to address this issue.

In Figure 1.1, an example is taken from the PersonaChat. In a conversation involving the persona of the second speaker (green one), a counterfactual sentence related to the persona is added to the last sentence of the second speaker in line with the task. In

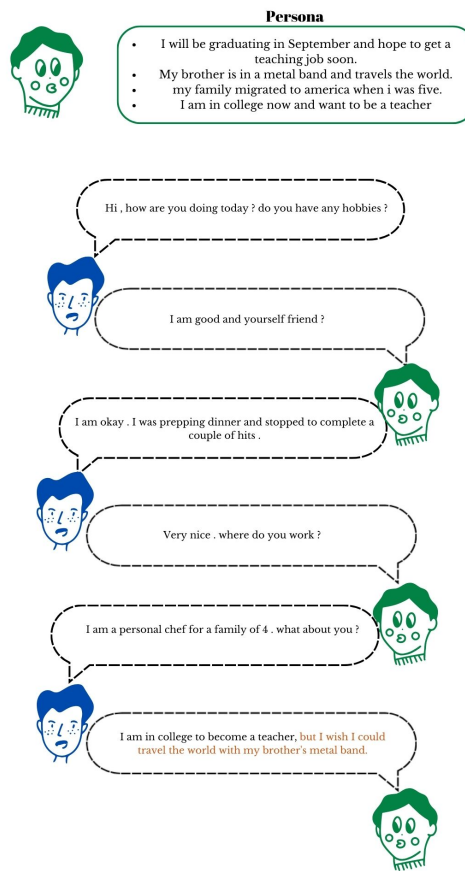


Figure 1.1 Dialogue with Counterfactual Sentence Added

this way, the aim is to increase consistency within the dialogue.

We argue that we can improve the consistency of personas in persona-based conversations by incorporating counterfactual sentences. We enhanced the PersonaChat dataset by adding counterfactual sentences generated based on the persona and dialogue history. Additionally, as the PersonaChat dataset is crowd-sourced, we made minimal changes to the dataset without altering the flow of the conversation. We evaluated the modified dataset through a successful metrics-driven study. The basis for evaluation was the Learning to Memorize Entailment and Discourse Relations (LMEDR) (R. Chen et al. 2023) study, and the revised dataset's results were subsequently evaluated based on this study.

Our main contributions can be summarized as follows:

- We show that complex tasks can be successfully performed with LLMs using the few-shot learning technique without fine-tuning.
- For each dialogue in the PersonaChat, a counterfactual sentence related to the persona was produced using the LLMs few-shot learning method. This work has been shown to be accomplished for different LLMs.
- The effect of counterfactual sentences in the augmented PersonaChat has been demonstrated experimentally.
- Revised PersonaChat dataset, augmented with counterfactual sentences, led to improved persona consistency in model-generated responses.

CHAPTER 2

BACKGROUND

2.1. Language Models

A language model is a probability distribution over sequences of words. When given a sequence of length m , it assigns a probability $P(w_1, w_2, \dots, w_m)$ to the entire sequence (Manning, Raghavan, and Schütze 2008). Language models are machine learning models trained on text data that try to predict the sentence or the next most appropriate word based on the context of the sentence or text. Language models are able to produce human-like sentences using Natural Language Processing (NLP) techniques, which have two categories: statistical models and models based on deep neural networks. Statistical models model and process the text using statistical methods while trying to comprehend a language structure by analyzing word order, the model, and probabilities in a language.

An n-gram model is a probabilistic language model that predicts the next item in a sequence based on the preceding $n - 1$ items. In the field of natural language processing, an n-gram model can be utilized to predict the next word in a sequence of words. (Daniel Jurafsky 2023).

For more complex and large-scale tasks, n-gram language models may be insufficient. Based on this problem has led to the development of deep learning-based language models.

Neural Language Models are deep learning-based language models. These achieve preferable results in word prediction, can process lengthier histories, and generalize better over the contexts of comparable words than n-gram models (Daniel Jurafsky 2023). They use neural networks to understand and process texts, with many of these models utilizing the transformer neural network architecture as shown in Figure 2.1 (Vaswani et al. 2017).

Models such as BERT, GPT, T5, and RoBERTa are examples of these language models.

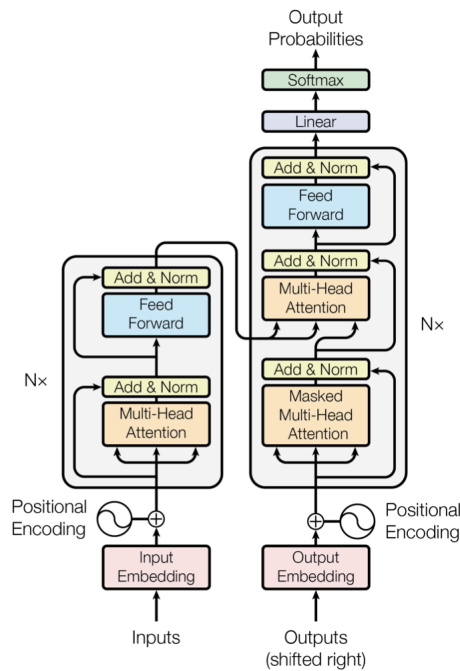


Figure 2.1 The Transformer - model architecture (Vaswani et al. 2017)

Transformers learn the structure and patterns of a defined language by following the relationships between words in a sentence, thereby understanding context and meaning. Encoder and Decoder are the basic components of NLP models and they play important roles, especially in the transformer architecture. In this architecture, the encoder processes the input sequence by capturing the relationships between elements to create a set of meaningful representations. The decoder generates contextual representations and produces the final output using the encoded vector.

The Transformer architecture is built upon the attention mechanism (Bahdanau, Cho, and Bengio 2016). The attention mechanism focuses on specific elements in the input sequence, allowing the context of those elements to be learned. Attention is available in both the encoder and decoder sections. Self-attention (Vaswani et al. 2017), on the other hand, enables learning the relationship of a word in a sentence or text with different words

in it. Likewise, self-attention can be found in both the encoder and decoder. Due to this mechanism, longer contexts and distant relationships can be learned.

In self-attention, a sequence of vectors enters as input, and a sequence of vectors comes out as output. Self-attention takes the weighted average of all input vectors to produce the output vector, and the simplest method to do this is the dot product called The Scaled Dot-Product Attention (Vaswani et al. 2017). Each input consists of queries with keys of size (d_k) and values of size (d_v) . The dot product of the query is computed using the keys. Each result is divided by the square root of (d_k) . The softmax function is then applied to obtain the weights of the values. In Figure 2.2, Scaled Dot-Product Attention is shown on the left side. The operations performed can be formulated as following equation 2.1 (Vaswani et al. 2017):

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.1)$$

The multi-head attention mechanism is an extended version of the self-attention mechanism and uses more than one self-attention head. Each attention header projects query, key, and value vectors with different weight matrices and calculates an independent scaled dot-product attention. This enables the model to learn different relationships and dependencies in parallel, as shown in the left image in Figure 2.2. The formula is represented by the following equation 2.2 (Vaswani et al. 2017):

$$MultiHead(Q, K, V) = Concat(head_1 \dots head_n)W^O \quad (2.2)$$

where $head_i = Attention(QW_Q^i, KW_K^i, VW_V^i)$

Published by Google, Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al. 2019) is a transformer that has been previously trained with a

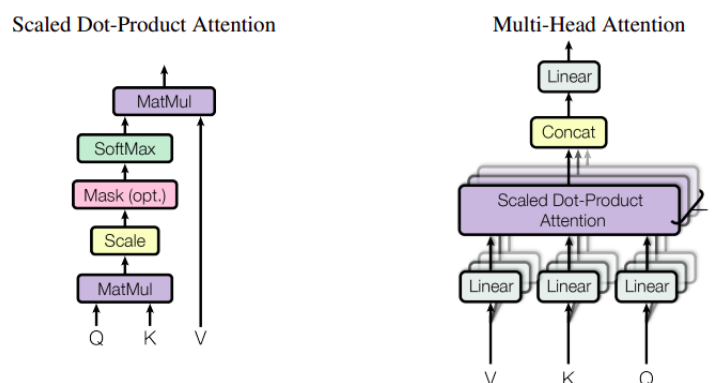


Figure 2.2 Scaled Dot-Product Attention on the left. Multi-Head Attention on the right (Vaswani et al. 2017)

substantial amount of text and consists only of encoder layers. This structure allows it to process the entire input context using bidirectional attention mechanisms simultaneously. BERT eliminates the one-way restriction compared to standard transformer models that use the Masking Language Modelling (MLM) and Next Sentence Prediction (NSP) method and provides an approach to word contexts regardless of right or left direction. In the MLM method, it masks some tokens in the input. Using MLM, BERT predicts the masked word based on the context around these tokens. This allows us to pre-train a deep bidirectional Transformer. NSP, or Next Sentence Prediction, is another one of BERT's pre-training tasks. Thanks to NSP, BERT not only learns the relationships between words but also understands the sequence of sentences. During the pre-training phase, BERT takes pairs of sentences as input with a special separator token ([SEP]) between them. The NSP task is a binary classification task, and BERT learns to predict whether the given sentences are consecutive or not.

BERT has two stages: pre-training and fine-tuning. In the pre-training stage, the trainee is trained with unlabeled data for different tasks. In the other fine-tuning stage, the pre-trained model optimizes the model's parameters for the desired task by fine-tuning it (Devlin et al. 2019).

GPT models are an example of Transformer decoders whose models are called “decoder-only” and contain only a decoder in their architecture. GPT is often pre-trained on large unlabeled -this process is called” generative pre-training”- text data and can then be used in various natural language processing tasks. Through this methodology, the model acquires general language abilities. After that, the model is fine-tuned for specific tasks. The general grammar learned during the generative pre-training phase enables better performance in particular tasks (Radford et al. 2018).

Table 2.1 GPT Language Models Comparison

Language Model	Number of Parameters	Number of Layers
GPT-1 (Radford et al. 2018)	117M	12
GPT-2 XL (Radford et al. 2019)	1.5B	48
GPT-3 Davinci (Brown et al. 2020)	175B	96

The GPT-1 model has been shown to outperform transformed-based language models in many NLP tasks with unsupervised pre-training and supervised fine-tuning (Radford et al. 2018). In GPT-2, which was created by improving GPT-1, the concepts of zero-shot, one-shot, and few-shot learning techniques were introduced (Radford et al. 2019). The other version of GPT-1 and GPT-2 is GPT-3, which has been shown to achieve state-of-the-art performance with few-shot learning in many various NLP tasks (Brown et al. 2020).

T5 (Text-to-Text Transfer Transformer) is a language model developed by Google Research in 2019 that uses a text-to-text approach. The architecture of T5 consists of the encoder-decoder transformer structure. T5 is trained with a massive textual corpus called “Colossal Clean Crawled Corpus” (C4) (Raffel et al. 2019). Text-to-Text framework was introduced with the T5 model. In this structure, the input is fed with text, and the results are produced as text, as seen in Figure. 2.3.

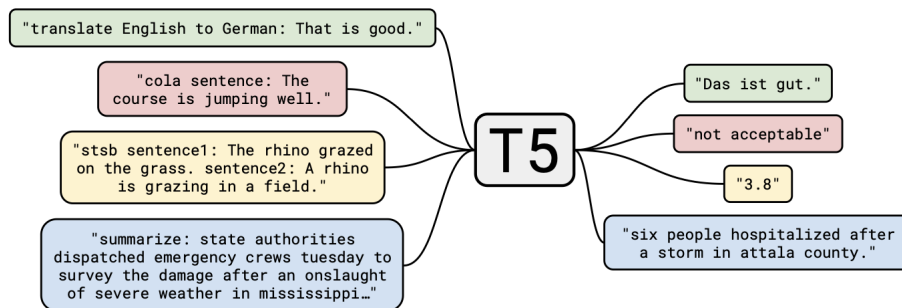


Figure 2.3 A Diagram of Text-to-Text Framework (Raffel et al. 2019)

2.1.1. Large Language Models (LLMs)

The development of language models has gained great momentum over time and reached gigantic sizes called "large." With the release of the GPT-3 model (Brown et al. 2020), which has 175 billion parameters, from the popular GPT series, the definition of "large" became popular and began to take place in our lives. After that, LLMs have become an important milestone in NLP. Also, LLMs leverage deep learning frameworks and large textual data.

Their basis is the transformer architecture. During the training process, they learn language contexts, sentence structures, and general language rules by using large language datasets (written source: articles, news, etc.) so they learn to predict the next word based on previous sentence contexts.

LLMs can be used for many tasks that humans can perform. These are text generation, content summarizing, AI bots, code generation, language translation, etc., also shown in Figure 2.4. Many known LLMs exist, such as OpenAI's GPT models and ChatGPT, Meta's Llama model, and Google's BERT model (IBM, n.d.).

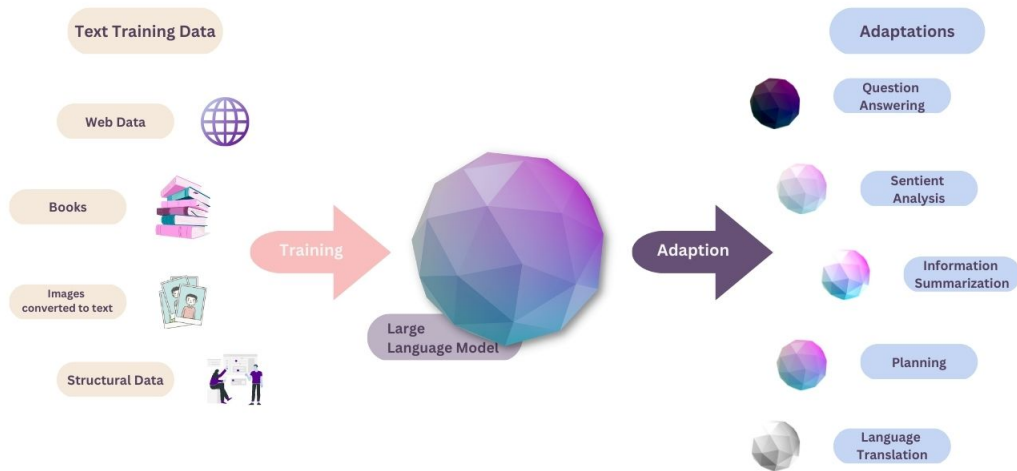


Figure 2.4 Real-World Applications of LLMs (Rachel 2022)

2.1.2. Pre-training and Fine-tuning

Pre-training involves training large language models to learn basic grammar and general structure using a large corpus. Additionally, the language model learns word relationships and contextual meanings during the training process.

Pre-training generally occurs without using labeled data, meaning it is done in an unsupervised or self-supervised manner. As explained in the BERT model, MLM (Masked Language Model) and NSP (Next Sentence Prediction) are used to train the model in the pre-training phase.

Fine-tuning involves retraining the model for a specific NLP task, incorporating knowledge gained during the pre-training phase. During this phase, the model is trained in a supervised manner using a smaller, task-specific dataset. Fine-tuning helps improve the model's performance on a particular task or domain by adjusting its weights to better fit the data.

2.1.3. Prompting

Prompting is the process of giving initial instructions or some input to a language model to perform a desired task. The way it is used in NLP is that instructions are provided in the form of text, which helps to produce the desired output. Due to prompting, desired outputs can be obtained consistently and accurately, but prompting results may vary depending on many different factors.

Prompts are very sensitive to the words and structure of the prompt. A slight change in the prompt changes the output. Besides, models may have problems creating the desired output in prompts that are ambiguous or contain insufficient context. Furthermore, the models may not provide the correct output in an assignment that is more complex and requires unique resources. Prompts should be given clearly and in detail, and the results should be evaluated considering the abovementioned situations.

Prompts can be given to language models in many ways, but the prompt must be well-designed to get the desired output. The prompt may contain information, context, a question, or many examples.

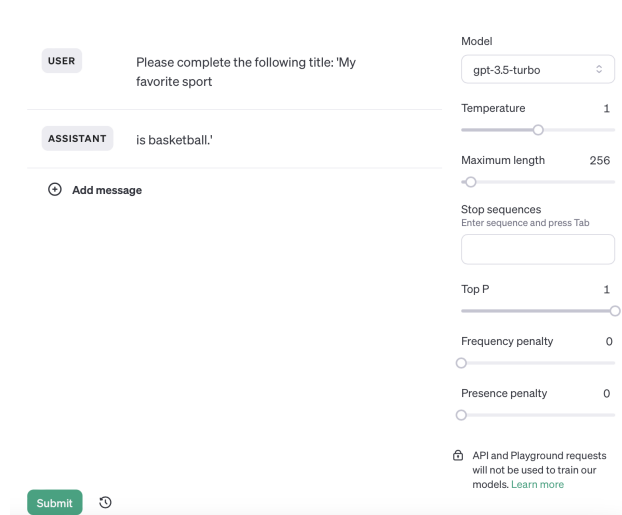


Figure 2.5 Prompt Example (OpenAI 2024)

Figure 2.5 shows the system, user, and assistant sections. The system prompt is not required but will be the road map for the assistant's response. In the example given, a prompt is provided directly by the User (Prompt Engineering Guide 2024).

Re-training language models incurs a significant cost. Some techniques have been developed to solve this problem. In the zero-shot learning method, the language model is given no examples. A response is expected to be generated directly from the language model.

One-shot learning is another example learning technique. This learning technique allows the language model to label new data on a single example. Also, few-shot learning is one of these techniques, and it also has several examples integrated into the language model. By using this, the language model has the ability to produce similar answers by using these examples in subsequent questions.

2.1.3.1 Zero-shot Prompting

In a machine learning scenario known as "zero-shot learning" (ZSL), an AI model is trained to identify and classify concepts or objects without having previously seen any examples of those categories or concepts (Bergmann 2024).

Zero-shot prompting is a variant of zero-shot learning. It is a method that enables a language model to perform a given task without needing any prior labeled data. In other words, it allows the given task to be realized using the information the model has previously learned. If you do not have sufficient labeled data or if you lack the necessary resources or time to collect labeled data, the model can be trained using the previously mentioned method. Nevertheless, it is likely that issues may arise. First of all, a stable result may not be obtained. Additionally, the desired answer may not be obtained in complex tasks.

For instance, as shown in Figure 2.6, by examining the sentence structures of a text, we can ask whether it is a news, blog, article, etc. Using its pre-trained knowledge, the model can determine which class it belongs to and produce an output. The introductory

text is examined. It establishes a relationship with predetermined classes and completes the classification.

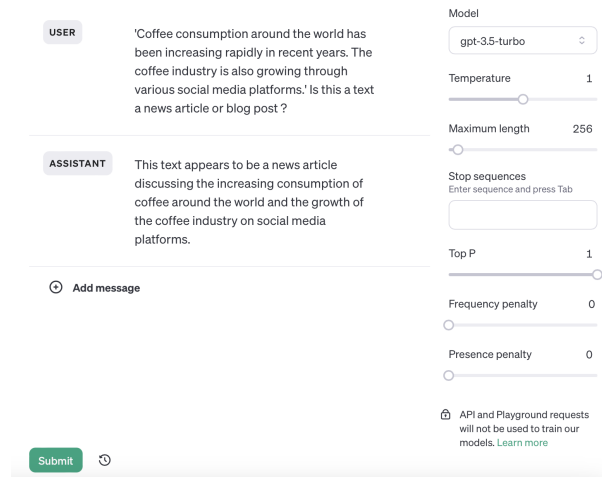


Figure 2.6 Zero-shot Prompting Example (OpenAI 2024)

2.1.3.2 Few-shot Prompting

Few-shot prompting is a paradigm in which a language model has a limited number of labeled data to perform a given task. Labeled data provided to the model are examples explicitly given for the task, which thus help the model learn the given task. The model works effectively with few-shot learning and quickly adapts to new classes. It also provides guidance for creating the output format with a few examples.

If Figure 2.6 is designed for few-shot prompting, several text documents for each category can be given as an example of the model. In the prompt, we can provide a few news articles for the news category and a few blog examples for blog posts. Accordingly, the model will make the classification more accurate for a given sample text by using labeled examples. The examples given here should be targeted, and there should be more examples for more complex tasks.

2.2. Applications of Language Models

2.2.1. Natural Language Understanding (NLU)

NLU is a subbranch of NLP and is the process by which computer systems understand and interpret natural language. NLU can be divided into three phases. First, the text goes through a preprocessing phase, dividing the incoming input into tokens. This involves standardizing lowercase/uppercase letters, special characters, and punctuation marks. Frequently used words that do not add meaning, such as conjunctions like "and" and "or," are removed. In the next stage, called Syntactic Analysis, the grammatical structures of the sentences are extracted. The text's words are categorized into nouns, adjectives, verbs, and adverbs. In the final stage, which is Semantic Analysis, the meanings of the words are determined based on the context (Word Sense Disambiguation). Notable names, such as people and places, are detected through Named Entity Recognition (NER). Furthermore, references that refer to the same entity in the text are identified.

One of the essential early works that can be considered the precursor of NLU is ELIZA (Weizenbaum 1966). ELIZA is a computer program developed by Joseph Weizenbaum at MIT in the mid-1960s. ELIZA is considered a significant milestone in the field of natural language processing. It has a place in history as one of the first NLU programs.

All recent work uses deep neural networks that are pre-trained with large amounts of unlabeled text data and then fine-tuned for downstream tasks. With billions of parameters, LLMs have harnessed the power of deep learning and extensive data to achieve remarkable performance in NLU and NLG tasks. Numerous studies have used transformers such as GPT and BERT for natural language understanding (NLU) research.(C.-S. Wu et al. 2020) (Meng et al. 2022) (X. Liu et al. 2023).

2.2.2. Natural Language Generation (NLG)

NLG is a sub-field of artificial intelligence that enables it to generate human-like natural language text or speech. NLG is a part of the NLP field and is used in many areas (presenting information, creating reports, producing answers in dialogues, etc.). NLG can be used to generate both open-ended answers and closed-ended answers in dialog systems. Open-ended answers involve creativity, and answers are generated broadly and freely. Closed-ended answers are answers to questions that require a specific and limited answer. These answers are used to provide transparent information.

The summarization task is one of the crucial NLG tasks. NLG algorithms extract critical information from the given text, identify essential sentences, and use this information to create a summary that reflects the original text. Once fine-tuned, NLG models can successfully perform many of these tasks. To illustrate, after training on multilingual datasets, NLG models can translate given text from one language to another, preserving its meaning and context.

The preliminary work in the field of NLG is considered to be the system called SHRDLU (Winograd 1972), developed by Terry Winograd in 1972. It is a system that understands commands in natural language and carries out these commands. NLU and NLG were used together in this study and thus became important representatives of early NLG research.

2.2.3. Natural Language Inference (NLI)

NLI is the process of determining the logical relationship of a piece of text (premises) with a sentence related to the text (hypothesis). NLI is crucial in semantic meaning analysis in sentences and contextual inference processes. These relationships generally consist of categories of entailment, neutral, and contradiction. Entailment indicates that the hypothesis can be validated as a premise. Contradiction indicates that the

hypothesis contradicts the premises. Neutral means that the hypothesis and the premises are neither rejected nor confirmed. They are illustrated with examples in the table 2.2.

Table 2.2 Examples of the Entailment, Neutral and Contradiction

Premises	Label	Hypothesis
A man examines green T-shirts.	contradiction	Man travels by bus.
Sisters laughing in the park.	neutral	Lots of children in the park.
The man was walking on the beach with his dog.	entailment	The man has a dog.

The Stanford Natural Language Inference (SNLI) dataset (Bowman et al. 2015) was published by Stanford to test NLI tasks and is used as a standard benchmark in the NLI field. MultiNLI (Multi-Genre Natural Language Inference) (Williams, Nangia, and Bowman 2018) is designed according to the SNLI dataset scheme, but it is a dataset that offers a wider variety of text.

NLI is used in personalized dialogue systems to generate responses based on the user’s identity, past communications, or preferences. Thus, with NLI, more natural and meaningful answers are produced with sentence context and persona. To illustrate, in a scenario where the user is vegan, vegan recipes can be suggested when the user asks what they can make for dinner tonight. In a scenario where the user likes action movies, action movie recommendations can be given when the user asks what they can watch.

2.2.4. Dialogue Systems

Dialogue is a communication process between two or more people, where feelings, thoughts, and ideas are shared between the parties. People use face-to-face or indirect (phone, e-mail, etc.) dialogues in their daily lives to share information and establish social bonds or social relations.

With the advancement of technology, human-computer interactions have increased. In this context, systems that can communicate in natural language are needed. Dialogue systems were born to meet this need. Dialog systems are systems that can communicate with people in natural language. It generates appropriate responses by understanding

users' natural language information.

Dialogue systems are generally divided into two groups. The first one, task-oriented dialogue systems, tries to solve specific problems in the given domain. Examples of these include: Restaurant Reservation Systems, Flight Booking Systems, Customer Support Chatbots. Compared to task-oriented dialogue systems, open-domain dialogue systems try to chat with users regardless of task and domain. Applications such as ChatGPT and Siri can be given as examples of dialogue systems in this group (Ni et al. 2021).

Task-oriented dialogue systems have traditionally consisted of four modules and have followed a pipeline structure. These modules are Natural Language Understanding (NLU), Dialogue State Tracking (DST), Policy Learning, and Natural Language Generation (NLG) (Ni et al. 2021).

The NLU module processes the user's message to extract meaningful information. It categorizes the user message into semantic slots, domain, and user intention. The DST module adjusts the dialogue state using the user's previous message and past dialog information. The Policy Learning module decides the next step based on the updated dialogue status and determines the action to be taken in line with the specified goal. Finally, the NLG module translates the selected action into a natural language response.

Open-domain dialogue systems are typically classified into three groups. Generative systems produce responses by utilizing the conversation history and user message through sequence-to-sequence models. Retrieval-based systems attempt to find the most appropriate answer from a given set of answers. Ensemble systems provide more effective dialogues by integrating the advantages of both methods (Zhu et al. 2018) (Y. Song et al. 2016) (Qiu et al. 2017) (Serban et al. 2017).

2.2.4.1 Personalized Dialogue Systems

Unlike classical dialogue systems, creating personalized dialogue is the process of interacting more personally and effectively by taking into account information such as the user's interests, hobbies, and preferences. During the conversation, the user will be

more interested in the conversation that he/she will experience with a chatbot that has a personality rather than a chatbot that has no personality. With human-like answers, there will be an increase in user experience and the time the user spends talking.

Thus, using personalized dialogues can provide advantages in many areas. In the field of education and personal development, appropriate learning content can be offered according to the learner's learning style or interests. In the field of customer service or marketing, it can provide suggestions to the user based on the customer's past shopping experiences and preferences.

A sufficient amount of personal data is needed to establish such dialog systems. There may be access restrictions for this information. In order to overcome this problem, a solution must be developed to use anonymous data or to enable personal information, preferences, and interests to be randomly generated and assigned to individuals, enabling them to talk. Mostly, anonymous data was used in the studies to solve these problems. Moreover, in these studies, random personal information was generated, and anonymous users were given this information and asked to chat (Zhang et al. 2018) (Li et al. 2016).

Early work in this area focused on user modeling and integrating personal information into dialogue (Lieberman 1995). In subsequent studies, more sophisticated and dynamic dialogue systems were developed using deep learning and big data techniques (Zhang et al. 2018)(Li et al. 2016).

2.3. Counterfactual Statements

Counterfactual perspective is a frequently used phrase in conversation. We use it to express what might happen or what we would like in an event that did not occur or in scenarios that we desired event. We can not expect this perspective from chatbots or language models because it is inherently humane and reflects personality traits (Pawar 2022).

It adds richness and flexibility to natural language in the dialogue creation process.

In this way, the sentences created are more natural, attractive to the user, and impressive. It also gives information about the user's character. Such as,

- "If I had caught the train, I would have arrived at the meeting early."
- "I wish I hadn't wasted so much time and had studied harder for the exam."

The sentences above were created with counterfactual thinking, with possible scenarios in case the event occurs and if it does not occur.

Counterfactual sentence detection is the process of detecting scenarios indicating that a specific event or statement could be different from the actual situation. In the field of NLP, Counterfactual sentence detection is the automatic detection of possible scenarios or alternative situations in texts. Although the concept of counterfactuals is mainly located in social sciences and philosophy, its integration into studies in the field of NLP started in the early 2000s. "Counterfactual thinking" (Roese 1997), one of the first studies in psychology, formed the basis of counterfactual sentence detection methods in the field of NLP. One of the studies proposed the first method to detect counterfactual sentences in texts: "Recognizing Counterfactual Sentences with Fact Probing" (Son, Lee, and Park 2014). Fact-checking and language modeling techniques are used in this study.

CHAPTER 3

RELATED WORK

Studies on persona information integrated into dialogue agents have improved in recent years. Especially with the appearance of datasets such as PersonaChat on this subject, many works have been published to solve the problems encountered in recent works. Zhang's work in 2018 introduced PersonaChat and studied two class models for the next utterance prediction task using this dataset: the Ranking model and the Generative model (Zhang et al. 2018). Ranking models try to predict the gold response as a response, separating out candidate sentences. The gold response is within the candidate sentences and generally represents real human answers. Generative models generate word-by-word responses using dialogue history and persona. TransferTransfo(2019) (Wolf et al. 2019) fine-tuned the pre-trained GPT2 model on the dataset and evaluated the results.

Natural language inference (NLI) has recently been frequently used to solve consistency problems. NLI makes a classification prediction between the premise and hypothesis as entailment, neutral, or contradictory. Dialogue NLI research was carried out in 2019 by Welleck et al. (Welleck et al. 2019). It gained momentum following the publication. Welleck et al. constructed the pairs of Dialogue NLI dataset using the PersonaChat dataset. Another study in this field aimed to provide consistency between response and agent by training the BERT over BERT (BoB) (H. Song et al. 2021) decoder with MultiNLI (MNLI) (Williams, Nangia, and Bowman 2018) dataset. BoB comprises a Bert-based encoder and two Bert-based decoders. One decoder is for response generation, while the other is for consistency understanding. Persona and Context Fusion (PCF) (Wang et al. 2023) aimed to ensure coherence by adding an extra NLI module compared to BoB. This way, they are targeted to ensure coherence between query and response. LMEDR (R. Chen et al. 2023) memorizes entailment and discourse relations using the memory networks structure. The

proposed approach discussed the problem of persona and dialogue context consistency in long dialogues, and the results were promising for future works. The SimOAP (Zhou et al. 2023) work has shown that high probability responses are not always the best responses than low probability responses. During the post-filtering process in the SimOAP work, the coherence of candidate responses is controlled with the TF-IDF method. Then, the selected responses are passed through the pre-trained NLI model to determine consistent responses with persona.

Cao et al. (Cao et al. 2022) addressed the data scarcity problem and manipulated the data with model-agnostic methods. In this work, only persona-related query and response dialogs were selected, all personas that were not entailed with response were deleted, and the number of dialogs and personas was increased using BERT, GPT2, and back translation techniques. Pre-trained NLI models were used to ensure the consistency of the augmented personas and responses and the coherent dialog history and response.

Persona-Adaptive Attention (Huang et al. 2023) work designed a dynamically regulatable attention mechanism to provide the context of the responses to the persona or context. Since not all responses will be related to the persona and sometimes need to be given depending on the context of the dialogue, the attention mechanism was used to provide this structure dynamically.

Ou et al. (Ou et al. 2022) tackled the issue of limited training data for dialogue systems. They proposed the Counterfactual Data Augmentation (CAPT) method to generate diverse responses with varying semantics based on the given dialogue history.

Synthetic dialogue datasets have been generated to enhance the learning capabilities of large language models (LLMs) through prompting. Lee et al. (Lee et al. 2022) presented a synthetic dialogue dataset called PERSONACHATGEN that was created with the learning capabilities of GPT-3. Chen et al. (M. Chen et al. 2023) explained that using prompting techniques in PLMs can help create more consistent and context-appropriate dialogue sets synthetically in the models.

In this study, the dialogues in the PersonaChat dataset were augmented by applying

prompting techniques to LLMs, thus aiming to increase persona consistency. Proposed by Ou et al. (Ou et al. 2022), the method works similarly to the Counterfactual Data Augmentation method used in the study. The answers in the dataset were augmented based on persona and context with a Counterfactual perspective. Counterfactual sentences were produced using the prompting technique and learning capabilities of PLMs. In the evaluation step, the LMEDR (R. Chen et al. 2023) study was used as the basis, and the results of our work were evaluated in this study. Similar studies and their summary are included in Table 3.1.

Table 3.1 Related Work Comparison

Study	Dataset	Method Details
BoB (H. Song et al. 2021)	PersonaChat and MNLI	Utilized BERT for consistent persona-based dialogue, enhancing response coherence with NLI integration.
D ³ (Cao <i>Et al.</i> 2022)	PersonaChat and DNLI	Implemented data distillation, diversification, and curriculum learning to enhance dialogue consistency and coherence.
CAPT (Ou et al. 2022)	Chinese Weibo corpus	Utilized counterfactual inference to generate diverse and high-quality dialogue responses by substituting observed reply perspectives with semantically different alternatives.
PCF (Wang et al. 2023)	PersonaChat and MNLI	Similar framework to the BoB study, but an additional NLI module was added to increase coherence.
LMEDR (M. Chen et al. 2023)	PersonaChat and DNLI	Used memory networks structure and discussed persona and dialogue context consistency.
SimOAP (Zhou et al. 2023)	PersonaChat	Generated and post-filtered response candidates for coherence using TF-IDF and for consistency with a pre-trained NLI model.
Persona-Adaptive Attention (Huang et al. 2023)	PersonaChat	Employed a dynamically regulatable attention mechanism to adaptively balance persona relevance and conversational context in dialogue responses.
Our Study	Augmented PersonaChat and Augmented DNLI	Augmented the PersonaChat dataset with counterfactual sentences and evaluated the enhancements using the LMEDR study.

CHAPTER 4

DATASET

PersonaChat (Zhang et al. 2018) is one of the most frequently used datasets in personalized dialogue generation tasks. This data set is utilized during the training and evaluation steps for chatbots engaging in conversations and incorporates various personality traits. Works completed on this dataset are also explicitly outlined in Chapter 3.

Moreover, if the dataset details are analyzed, then information reflecting the personality (hobbies, preferences, etc.) and the dialogue history of the two speakers is shown. Candidate answers are also included for each answer. Since this data set was crowd-sourced using Amazon Mechanical Turk, it was created by giving some instructions to people. At this stage, they were asked to get to know each other and reflect on their personality traits in their answers. Figure 4.1 shows how the dataset was created.

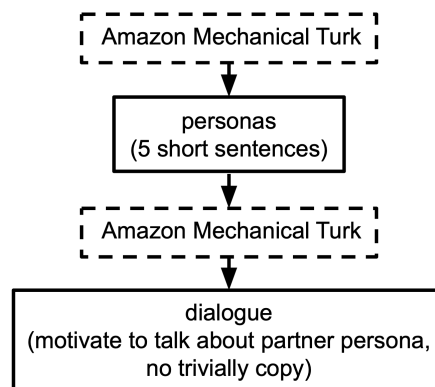


Figure 4.1 Creation Stages of PersonaChat Dataset (Princeton University 2020)

The dataset comprises an extensive dialogue network, distinguishing itself from other datasets regarding average utterance per dialogue and total number of dialogues. It

encompasses a wide range. Below are commonly used dialogue datasets and statistics in Table 4.1.

Table 4.1 Some Dialogue Datasets Statistics

Dataset	Source	Number of Dialogue	Utterance per Dialogue
PersonaChat (Zhang et al. 2018)	Crowd	10.9K	14.9
PersonalDialog (Zheng et al. 2019)	Weibo	20.8M	2.7
PersonaMinEdit (C. H. Wu et al. 2021)	Crowd	121.8K	2
FoCus (Jang et al. 2021)	Crowd	13.5K	11.3
BlendedSkillTalk (Jang et al. 2021)	Crowd	6.8K	11.2
MSC (Xu, Szlam, and Weston 2022)	Crowd	24K	12.5
IT-ConvAI2 (Y. Liu et al. 2022)	ConvAI2	1.6K	2
MPChat (Ahn et al. 2023)	Reddit	15K	2.8

Given the dataset format, conducting preprocessing stages is essential to extracting the dialogues directly. Each dialogue contains information about the persona, the speakers' sentences, and the candidates' answers. Candidate answers are used to evaluate performance in Ranked models. The gold response is the sentence used by the user in the actual dialogue, which is located last among the candidate's answers. In order to recreate the dialogues from the dataset, it is necessary to parse the persona information, the sentences of the first speaker, and the sentences of the speaker whose persona is given.

There is still work to be done on the PersonaChat dataset to improve consistency and coherence. These works can be explained as producing answers related to the persona (consistency) and ensuring the dialogue flow (coherence). The dataset can be enlarged to improve specified metrics, or the dialogues in the training dataset can be changed. Expanding the dataset can be costly because people create every dialog in the PersonaChat dataset, and human labor is costly. Reaching more Turkers will result in higher costs. Another method can be considered as changes to be made in the dialogues. We recommend changing the dialogs to make them more persona-consistent. In particular, we propose

adding counterfactual sentences to the dialogues in the train dataset to create more human-like generated answers.

CHAPTER 5

METHODOLOGY

Figure 5.1 shows the Project’s overall framework. Initially, counterfactual sentences are detected for the sentences of the user whose persona is given in the PersonaChat dataset. The cleaned dataset is saved after removing dialogues containing counterfactual sentences from the dataset. Later, a few dialogues are selected for few-shot learning from dialogues containing counterfactual sentences added to the dialogue manually or by the language model. After choosing the large language models, the designed system prompt, few-shot prompt, and input prompt are sent to LLM, and then revised dialogues with counterfactual sentences are recorded.

In the evaluation step, the framework in the LMEDR study was used. LMEDR has demonstrated state-of-the-art performance across many metrics (R. Chen et al. 2023). The revised PersonaChat train dataset was created after our framework, the Dialogue Natural Language Inference (DNLI) dataset containing the newly generated sentences was added to the LMEDR module, and a new model was trained. Metric calculations are made using the trained model PersonaChat test dataset. Figure 5.2 shows the evaluation steps in detail.

5.1. Counterfactual Sentences Detection in Dataset

The Bert-base Multilingual Uncased (Devlin et al. 2019) model was used to detect counterfactual sentences in the dataset dialogues. Amazon Multilingual Counterfactual Dataset (AMCD) (O’Neill et al. 2021) English was used as the model train and validation datasets. After training and validation, PersonaChat dataset dialogues were used as test datasets. Preprocessing operations were carried out to extract the dialogues in the PersonaChat dataset. The sentences of the user whose persona is given in the dialogues

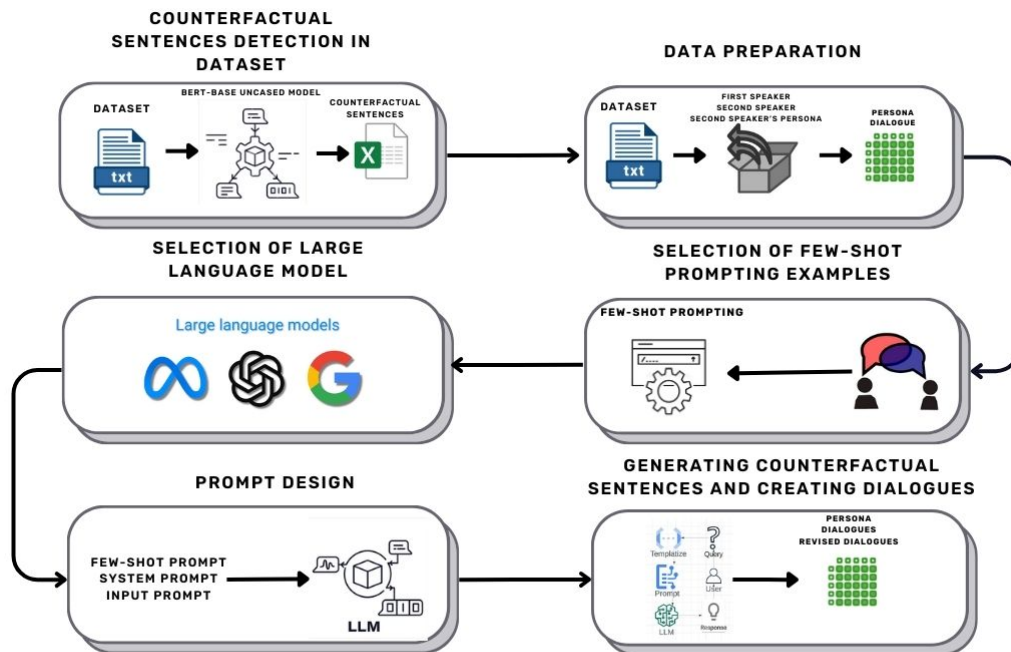


Figure 5.1 Project Workflow

in the PersonaChat dataset are extracted through preprocessing since we will not perform augmentation on the user whose persona is not given. A comprehensive file of counterfactual tagged sentences was prepared during the model testing process. Some detected sentences are given in Figure 5.3

5.2. Data Preparation

Upon examining the PersonaChat dataset, it becomes evident that it involves interactions between two speakers. Following the persona lines, the persona information of the person who speaks first in each line is not given. It may also be called "First Speaker". Speakers whose persona is given after the "First Speaker" will be called "Second Speaker".

The "|" separating the sentences are referred to as candidate sentences. The

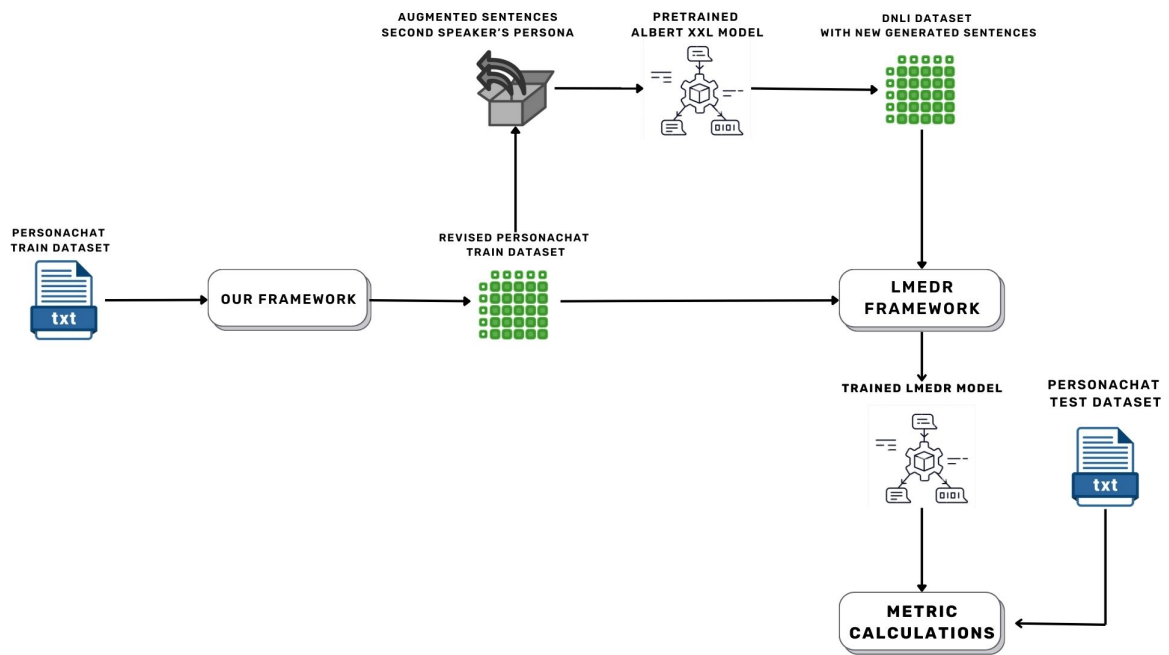


Figure 5.2 Evaluation Steps

sentence at the end of the candidate sentences is the actual answer given by the Second Speaker (Gold Response). Each dialogue will be labeled "First Speaker" and "Second Speaker" and included in the input prompt. Additionally, the "Second Speaker" persona information is kept the same. This information is sent to LLMs using the input prompt.

After the processing, the persona information of the second speaker and dialogues were arranged as in the example in Table 5.1.

5.3. Selection of Few-shot Prompting Examples

Prompting, as described in Background 2.1.3, is essential for getting the desired response. Zero-shot or few-shot learning techniques can be used to get the desired answer according to the task. This study used a few-shot learning technique for LLMs due to task complexity.

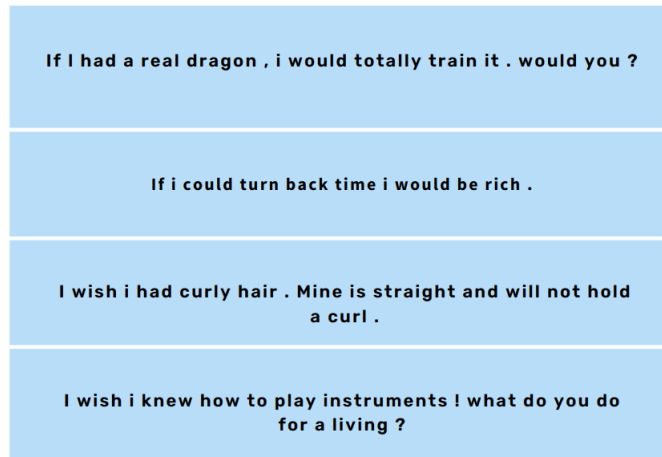


Figure 5.3 Some Counterfactual Sentences Detected From the PersonaChat

Since a counterfactual sentence will be added to the sentences in the few-shot examples after the original dialogue and persona information are provided, revised dialogues with a counterfactual sentence for "Second Speaker" should be given in the most appropriate place. The fact that this additional counterfactual statement is related to the user and the previous conversation will assist the LLMs in generating the intended response in various scenarios. When producing this example, dialogues with counterfactual sentences can be used from the original dataset or created using LLMs for the few-shot prompting. Figure 5.4 shows an example prepared for few-shot prompting. In the provided prompt, the AI is asked to generate a counterfactual sentence that is appropriate for the given dialogue and persona. A sample dialogue and person are provided as input. The output will show the AI's response, along with the counterfactual sentence that was generated before receiving the updated dialogue, and it will be mentioned that this counterfactual sentence is in line with the dialogue and persona. The input-output format in the few-shot prompt will serve as a sample template for the subsequent generated results. This approach will guide the AI in producing responses within this framework.

Table 5.1 An Example Line From the PersonaChat after Preprocessing

Persona	Dialogue
<p>I work as a computer programmer. I helped develop Facebook I make over 200,000 per year. I like to donate time and money to charity. I have two pit bulls.</p>	<p>First Speaker: hi, what are you up to? Second Speaker: what's up. First Speaker: I just finished cooking a rotisserie chicken with chimichurri sauce. You? Second Speaker: that sounds good. I am relaxing with my pups after a long day. First Speaker: how many do you have? I had a long day, too. That 9 5 hustle. Second Speaker: I have two pit bulls. What do you do? First Speaker: I have my own business, but I really want to open a bakery one day. You? Second Speaker: awesome. I am a programmer, and in my free time, I work at a soup kitchen. First Speaker: that is awesome! I volunteer at my local soup kitchen, too! Second Speaker: cool! Where is yours? First Speaker: in Virginia. I actually helped set it up in the area. Second Speaker: nice. Mine is in downtown Austin, next to the civic center.</p>

HUMAN

Generate a new dialog with generated counterfactual sentence. The speakers should speak as many times in the new dialogue you will create as the speakers speak in original dialogue.

First Speaker: hi , how are you doing today ? do you have any hobbies ?

Second Speaker: i am good and yourself friend ?

First Speaker: i am okay . i was prepping dinner and stopped to complete a couple of hits .

Second Speaker: very nice . where do you work ?

First Speaker: i am a personal chef for a family of 4 . what about you ?

Second Speaker: i am in college to become a teacher

First Speaker: oh , that is cool . the career highlight of my schooling was winning a spelling bee .

Second Speaker: was that in 3rd grade ?

First Speaker: it was in the 2nd grade . i missed the cut in the 3rd grade . haha .

Second Speaker: dang they do move on to 5 letter words then . rough .

First Speaker: i know , right ? that is why culinary school was where i ended up .

Second Speaker: make sure you spell the food right !

and

Second Speaker's Persona: i will be graduating in september and hope to get a teaching job soon. my brother is in a metal band and travels the world. my family migrated to america when i was five. i am in college now and want to be a teacher.

AI

Here is the revised dialogue appears below. The counterfactual sentence "i wish I could travel the world with my brother's metal band." was generated in relation to the second speaker's character and added to the second speaker's sentence. Preserved the dialogue flow and its length.

First Speaker: hi , how are you doing today ? do you have any hobbies ?

Second Speaker: i am good and yourself friend ?

First Speaker: i am okay . i was prepping dinner and stopped to complete a couple of hits .

Second Speaker: very nice . where do you work ?

First Speaker: i am a personal chef for a family of 4 . what about you ?

Second Speaker: i am in college to become a teacher, but i wish I could travel the world with my brother's metal band.

First Speaker: oh , that is cool . the career highlight of my schooling was winning a spelling bee .

Second Speaker: was that in 3rd grade ?

First Speaker: it was in the 2nd grade . i missed the cut in the 3rd grade . haha .

Second Speaker: dang they do move on to 5 letter words then . rough .

First Speaker: i know , right ? that is why culinary school was where i ended up .

Second Speaker: make sure you spell the food right !

Figure 5.4 One of the Few-Shot Prompting Examples

5.4. Selection of a Large Language Model

Many large language models can be used for this task. Llama, GPT, and Jurassic-1 models can be given as examples of these models. Working with several language models and considering specific parameters is crucial regarding performance comparison. A language model should be chosen depending on many factors, such as how fast the results will be obtained and the ability to solve complex problems. Another important point when selecting a large language model is how many parameters the model contains. A higher number of parameters may give higher performance, but the response time may increase accordingly.

Additionally, variables such as resources, memory, and cost are crucial. Models with more extensive parameters use more resources and memory. There are many open-source large language models available, such as Llama models. For example, some GPT models charge per token. This decision should also be based on the project's funding.

There are many usage parameters when using language models. Parameters such as temperature, top-p, and top-k should be determined according to your task. The temperature should decide whether or not you want more creative or deterministic results. While a higher temperature value produces more innovative results, a lower temperature value produces more predictable results. The Top-p value, also known as nucleus sampling, controls the cumulative probability of the generated tokens, ensuring that it produces tokens until this value exceeds the selected p-value. The Top-k parameter takes the most probable k tokens in each production.

GPT 3.5 Turbo (Brown et al. 2020) and Llama 2 13B (Touvron et al. 2023) models were used in this study. The temperature value was 0.7 for the GPT 3.5 Turbo and 0.6 for the Llama 2 13B model. Furthermore, Top-p and Top-k default values were selected as 0.95 and 40, respectively.

5.5. Prompt Design

Creating prompts to get what we want from LLMs has become essential. We can obtain the answer from LLMs if we provide specific examples of our objective. Counterfactual sentences can be added to the dialogues on the PersonaChat dataset, such as the examples mentioned in few-shot prompting. However, additional counterfactual sentences generated will not be added to the dialogues containing counterfactual sentences identified in Methodology 5.1 in the original PersonaChat dataset. If counterfactual sentences are added to these sentences again, the dialogue will be created containing mainly counterfactual sentences, and the naturalness will be removed.

Three separate prompt designs should be made. The System Prompt describes what we want from the language model. The input prompt will contain Dialog and Persona information specific to this task. A few-shot prompt will be created to get the desired output, consisting of several examples, as explained in the Methodology 5.3 section.

The system prompt instructions to follow are given below:

1. Identify every utterance of the second speaker.
2. Consider the character of the second speaker.
3. Generate counterfactual sentences based on the second speaker's character and the first speaker's utterances.
4. Keep the rest of the dialogue unchanged.
5. Ensure that only one counterfactual sentence remains in the revised dialogue, placed appropriately, while others are removed.
6. The speakers in the new dialogue should speak the same number of times as they do in the original dialogue.

The LLMs must perform two essential tasks: Producing a counterfactual sentence about the persona and placing this sentence in the appropriate place in the dialogue.

5.6. Generating Counterfactual Sentences and Creating Dialogues

Once the prompts are established, the next crucial step is to generate the revised PersonaChat dataset. New dialogues should be made with as few changes as possible because humans created the PersonaChat dataset, so the dialogues should be slightly changed and not separated from the general structure. Also, the dialogue length should remain constant, and the conversation flow should stay the same. What is desired here is explained in detail in the prompts in the Methodology 5.5. Here, the purpose of giving counterfactual sentences to people whose persona is given is to create persona-consistent responses.

Additionally, if the length of the dialogue changes and the desired output is not received, the prompts will be sent again. The language models may sometimes provide delayed or incorrect answers, even if prompted multiple times. Therefore, the request for a new response will be ignored after five attempts.

5.7. Evaluation

The evaluation steps are visualized in Figure 5.2. Performance evaluations should be carried out once we receive the revised dialogues. The LMEDR (R. Chen et al. 2023), which shows state-of-the-art performance in metrics in personalized dialogue generation, was taken as a basis. This study utilizes the PersonaChat and DNLI datasets. The DNLI dataset (Welleck et al. 2019) comprises sentence pairs from the PersonaChat dataset and is labeled with the NLI model. Due to the augmentation process in the PersonaChat dataset, it is also necessary to update the DNLI dataset. ALBERT pre-trained model was used in the labeling process. A fine-tuned pre-trained ALBERT model was used in the labeling process for DNLI dataset and C-score calculation. Fine-tuned and Pre-trained stages are shown in Figure 5.5. The LMEDR model was trained with the revised PersonaChat and DNLI datasets, and the results were compared with those of the LMEDR.

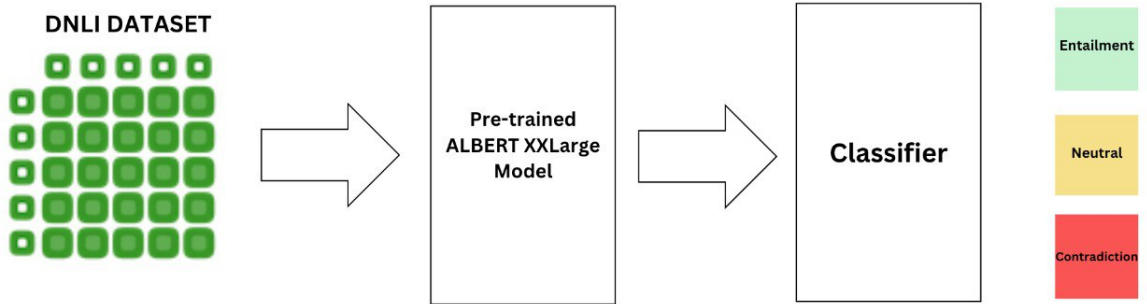


Figure 5.5 Fine-tuned Pre-trained NLI Model

Metric evaluations after the LMEDR study, trained with the revised dataset with Llama 2 13B and GPT 3.5 Turbo models, are made automatically using the ParlAI (Miller et al. 2018) module. ParlAI is an open-source software platform developed to test dialog models developed by Facebook AI Research. Metrics such as F1, Perplexity (PPL), and Hits@1 were calculated using the ParlAI framework. The F1 score is a crucial metric in evaluating personalized dialogue models that represent the word-level harmonic mean of precision and recall. Calculating the negative log probability of the predicted golden response is called PPL. Hits@1 is the probability that the golden response ranks the highest among the 20 candidate responses (Zhang et al. 2018). The model selects the best answer from the candidate answers, resulting in a score of 1 if the model chooses the correct response (golden response); otherwise, 0, as shown below equation 5.1:

$$Hits@1 = \frac{1}{M} \sum_{i=1}^M 1 \{ \hat{y}_i = y_i \} \quad (5.1)$$

where

M total test samples.

\hat{y}_i represents of model prediction response.

y_i represents the golden response.

These metrics provide a comprehensive analysis to evaluate the performance of the model. Another metric used to assess the performance of the personalized dialogue model is the C-score (Madotto et al. 2019). That is used to understand the consistency of the answer given with the persona. Its formula is shown below in equation 5.2 and 5.3.

$$\text{NLI}(r, p_j) = \begin{cases} 1 & \text{if } r \text{ entails } p_j \\ 0 & \text{if } r \text{ is independent to } p_j \\ -1 & \text{if } r \text{ contradicts } p_j \end{cases} \quad (5.2)$$

$$\text{C-score} = \frac{1}{M} \sum_j^m \text{NLI}(r, p_j) \quad (5.3)$$

where

r given response.

M total test samples.

m total number of persona sentences.

$j = 1, 2, \dots, m$.

p_j j^{th} persona sentence.

CHAPTER 6

EXPERIMENTS

6.1. Implementation Details

In our framework, illustrated in Figure 5.1, the initial step involves identifying the counterfactual sentences within the PersonaChat training dataset. Preprocessing processes were performed to extract the dialogues in the PersonaChat dataset, and the sentences of the second speaker - whose persona was given - were saved. Bert base Multilingual Uncased Model was used to detect whether these sentences contain counterfactual sentences. Initially, Bert base Multilingual Uncased Model was trained and validated with the AMCD dataset. During the training step, learning rate $1e-05$, training batch size 32, and max token length 256 were selected. This model was trained with T4 GPU in Google Colab (*Google Colaboratory*). After that, the trained model predicted counterfactual labels for the second speaker's sentences.

The reason for this detection phase is that 869 dialogues containing counterfactual sentences are not augmented again. Additionally, time inefficiency will be reduced because the augmentation will not be performed on those dialogues.

To prepare dialogues and personas to be sent to the input prompt, the PersonaChat train dataset must be preprocessed, and the persona information of the second speaker and dialogue sentences must be extracted. Since the data set also includes candidate answers, each personal information and dialogue must be parsed correctly in the given ".txt" format files.

Once the personas and dialogues needed for the input prompt are parsed and recorded, we must select the appropriate examples for few-shot learning. In this selection process, we can use dialogues that contain the sentences identified in the counterfactual

detection section. Alternatively, we can enhance dialogues by adding counterfactuals using LLMs. We used this technique's outputs as few-shot prompting examples.

As mentioned in Methodology 5.4, we implemented GPT 3.5 Turbo and Llama 2 13B models ¹ for counterfactual sentence augmentation. We used the LangChain ² framework for prompting the integration of both models. Also, Few-shot prompt, system prompt, and input prompt templates are designed using Langchain's Prompt Template methods. The LangChain is an open-source framework supported by LLMs where you can develop applications in the field of NLP (LangChain 2024).

After completing these tasks, prompts are sent for Llama and the GPT models, and the revised dialogs are received as an output. Prompts are formatted and sent for each dialogue that does not contain a counterfactual sentence. After the given answer is parsed, if there is a change in the number of dialogue utterances or the sent input dialogue and the received dialogue are the same, the prompt is sent again, and the answer is rechecked. This repetition process is performed five times. If the answers received are still not as desired, no augmentation is performed for that dialogue.

After the PersonaChat dataset is revised, it is evaluated in the LMEDR study based on it. The proposed LMEDR model is initialized with the BART-large. AdamW was used to optimize the model, and the initial learning rate was determined to be 8e-6. The batch size for training stage 1 was 64, and a batch size of 2 was used with a gradient accumulation of eight for the training stage. A beam search was used for dialogue generation with a maximum sequence length set to 50 (R. Chen et al. 2023).

In the evaluation part, F1, Hits@1, and PPL metrics are evaluated automatically due to the ParlAI framework. The C-score metric calculation is based on the formula given in Methodology 5.7. ALBERT XXLlarge model ³ was used as the NLI model that was trained with SNLI (Bowman et al. 2015), MNLI (Williams, Nangia, and Bowman 2018), FEVER-NLI (Nie et al. 2020), and ANLI (R1, R2, R3) (Nie et al. 2020) datasets.

1. <https://huggingface.co/TheBloke/Llama-2-13B-GGUF>

2. <https://api.python.langchain.com/>

3. https://huggingface.co/ynie/albert-xxlarge-v2-snli_mnli_fever_anli_R1_R2_R3-nli

The ALBERT XXLarge model labels the relationship between each persona and response. The C-score is calculated for each sample in the test set, and the metric score is calculated after dividing the C-score by the total number of samples in the test set.

6.2. Results and Discussion

The aim is to create a new PersonaChat dataset that contains more personalized sentences. For this purpose, the counterfactual sentence augmentation method has been preferred. Counterfactual sentences provide a different perspective. If counterfactual sentences related to the persona are added to the dialogues, characteristics such as naturalness and personalization will be enhanced.

Table 6.1 Statistics of Generated Counterfactual Sentences

Total Dialogues	8939
Total Detected Counterfactual Sentences	869
Generated Counterfactual Sentences by GPT 3.5	8070
Generated Counterfactual Sentences by Llama 2	5859

Table 6.2 Automatic Evaluation Results Comparison for Persona-based Dialogue Generation

Model	Hits@1	PPL	F1
Transfertransfo (Wolf et al. 2019)	82.1	17.51	19.09
P^2 BOT (Q. Liu et al. 2020)	81.9	15.12	19.77
LMEDR with Original PersonaChat	89.5	10.99	21.99
LMEDR with Augmented PersonaChat using GPT 3.5	86.4	11.58	21.27
LMEDR with Augmented PersonaChat using Llama 2	87.2	11.50	21.21

While examining the PersonaChat training dataset augmented with Llama 2, we observed that Llama 2 employs counterfactual sentences directly in the examples used in few-shot prompts in some revised dialogues. For example, a sentence like this is repeated in many dialogues: "If I could change one thing about my life, it would be to live in a city with a vibrant arts scene.". This shows that Llama 2 is limited in generating new counterfactual sentences creatively.

Additionally, not all dialogues could be augmented in Llama 2 because when repetitions are exceeded, the augmentation process stops, and the original dialogue is saved. For GPT 3.5, counterfactual sentences were generated for all dialogues except those already containing counterfactual sentences. These statistics are shown in Table 6.1. Therefore, Llama 2 is less affected by increasing counterfactual sentences, as seen in metrics other than the C-score and F1. Hits@1 and PPL values of the PersonaChat dataset augmented with Llama 2 approached the original LMEDR model. They gave better results than GPT 3.5 in these metrics, as shown in 6.2.

In addition, the robustness of the LMEDR study on augmented datasets was also observed. Although Llama 2 and GPT 3.5 results for F1, Hits@1, and PPL metrics are lower than the original study, better results were obtained than those of the P^2 BOT and Transfertransfo.

In LMEDR, the pre-trained BERT model was used to calculate the C-score. In this study, the pre-trained ALBERT XXLlarge model was used, and this model was fine-tuned with the augmented DNLI dataset. For this reason, C-score calculation was made again for the dialogues generated in the LMEDR study. Upon analyzing the results in Table 6.3, the PersonaChat dataset augmented with GPT 3.5 Turbo gave more persona-consistent results. In addition, comparisons were made between the pre-trained and fine-tuned models, as shown in Table 6.3. The values after fine-tuning increased significantly, and based on our observations, the fine-tuned model made label predictions better.

Table 6.3 C-score results calculated using pre-trained and pre-trained fine-tuned NLI models

	C-score using Pretrained	C-score using Fine-tuned
LMEDR with Original PersonaChat	0.1110	0.4462
LMEDR with Augmented PersonaChat using GPT 3.5	0.1345	0.4539
LMEDR with Augmented PersonaChat using Llama 2	0.1096	0.4453

CHAPTER 7

CONCLUSION

This study was conducted to increase persona consistency in personalized dialogue systems. The PersonaChat dataset was augmented using the data augmentation capabilities of LLMs. Additionally, since the PersonaChat dataset is crowd-sourced, the augmentation process was done in a way that would least affect the dataset. In the data mentioned earlier augmentation method, persona-consistent and coherent counterfactual sentences were added to the dialogues. LLMs learning capabilities were used to create these counterfactual sentences added to the dataset. GPT 3.5 Turbo and open-source Llama 2 13 B models were chosen to be compared in counterfactual sentence augmentation. After designing the few-shot, input prompt, and system prompts for the GPT 3.5 Turbo and Llama 2 13 B models, the augmentation method was applied for each dialogue in the PersonaChat dataset that did not contain a counterfactual sentence.

The PPL metric demonstrates the fluency of the produced sentence. Hits@1 and F1 metrics indirectly measure dialogue coherence as well. When evaluating the experimental results with metrics, although the results for F1, Hits@1, and PPL were lower than the LMEDR model, the dialogues enriched with GPT 3.5 showed more successful results in terms of C-score, so the answers produced are more persona-consistent. As in most studies, it is beneficial for the answers produced, especially for F1, Hits@1, and PPL, to undergo human evaluation because the answers produced with the golden response may differ. Still, the answers produced as dialogue coherence, persona consistency, and fluency evaluations may be compatible.

To compare the results with LMEDR, the ALBERT-XXLarge model was used as the NLI module for the C-Score calculation during the evaluation study, unlike the LMEDR study. At the same time, personalized dialogue generation comparisons of the

LMEDR models were also made for the Llama 2 and GPT 3.5 models. As a result of these studies, it was observed that the LMEDR study was also successful on augmented datasets.

7.1. Future Work

This study can also be a guide for future studies. There are still areas that can be improved to produce persona-consistency answers. With the development of LLMs' zero-shot and few-shot capabilities, successful studies continue even in complex tasks. This work can be developed in LLMs who have completed many tasks, such as GPT 4 (OpenAI et al. 2024) and Llama 3 (Meta 2024).

The results obtained can be assessed in terms of metrics such as coherence, consistency, and fluency through human evaluation, as automatic evaluations may not fully capture human perception.

REFERENCES

- Ahn, Jaewoo, Yeda Song, Sangdoo Yun, and Gunhee Kim. 2023. “MPCHAT: Towards Multimodal Persona-Grounded Conversation.” In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, edited by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, 3354–3377. Toronto, Canada: Association for Computational Linguistics, July. <https://doi.org/10.18653/v1/2023.acl-long.189>. <https://aclanthology.org/2023.acl-long.189>.
- Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2016. *Neural Machine Translation by Jointly Learning to Align and Translate*. arXiv: 1409.0473 [cs.CL].
- Bergmann, Dave. 2024. “What is zero-shot learning?” IBM. <https://www.ibm.com/topics/zero-shot-learning>.
- Bowman, Samuel R., Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. “A large annotated corpus for learning natural language inference.” In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 632–642.
- Brown, Tom B, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. “Language models are few-shot learners.” *arXiv preprint arXiv:2005.14165*.

Cao, Yu, Wei Bi, Meng Fang, Shuming Shi, and Dacheng Tao. 2022. “A Model-agnostic Data Manipulation Method for Persona-based Dialogue Generation.” In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, edited by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, 7984–8002. Dublin, Ireland: Association for Computational Linguistics, May. <https://doi.org/10.18653/v1/2022.acl-long.550>. <https://aclanthology.org/2022.acl-long.550>.

Chen, Maximillian, Alexandros Papangelis, Chenyang Tao, Seokhwan Kim, Andy Rosenbaum, Yang Liu, Zhou Yu, and Dilek Hakkani-Tur. 2023. “PLACES: Prompting Language Models for Social Conversation Synthesis.” In *Findings of the Association for Computational Linguistics: EACL 2023*, edited by Andreas Vlachos and Isabelle Augenstein, 844–868. Dubrovnik, Croatia: Association for Computational Linguistics, May. <https://doi.org/10.18653/v1/2023.findings-eacl.63>. <https://aclanthology.org/2023.findings-eacl.63>.

Chen, Ruijun, Jin Wang, Liang-Chih Yu, and Xuejie Zhang. 2023. *Learning to Memorize Entailment and Discourse Relations for Persona-Consistent Dialogues*. arXiv: 2301.04871 [cs.CL].

Daniel Jurafsky, James H. Martin. 2023. *Speech and Language Processing*. <https://web.stanford.edu/~jurafsky/slp3/3.pdf>.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. “BERT: Pre-training of deep bidirectional transformers for language understanding.” *arXiv preprint arXiv:1810.04805*.

Google Colaboratory. <https://colab.research.google.com/>. Google.

- Huang, Qiushi, Yu Zhang, Tom Ko, Xubo Liu, Bo Wu, Wenwu Wang, and H Tang. 2023. “Personalized Dialogue Generation with Persona-Adaptive Attention.” *Proceedings of the AAAI Conference on Artificial Intelligence* 37, no. 11 (June): 12916–12923. ISSN: 2159-5399. <https://doi.org/10.1609/aaai.v37i11.26518>. <http://dx.doi.org/10.1609/aaai.v37i11.26518>.
- IBM. n.d. “Large Language Models.” IBM. <https://www.ibm.com/topics/large-language-models>.
- Jang, Yoonna, Jungwoo Lim, Yuna Hur, Dongsuk Oh, Suhyune Son, Yeonsoo Lee, Dong-Hoon Shin, Seungryong Kim, and Heuseok Lim. 2021. “Call for Customized Conversation: Customized Conversation Grounding Persona and Knowledge.” *CoRR* abs/2112.08619. arXiv: 2112.08619. <https://arxiv.org/abs/2112.08619>.
- Kim, Donghyun, Youbin Ahn, Wongyu Kim, Chanhee Lee, Kyungchan Lee, Kyong-Ho Lee, Jeonguk Kim, Donghoon Shin, and Yeonsoo Lee. 2023. “Persona Expansion with Commonsense Knowledge for Diverse and Consistent Response Generation.” In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, edited by Andreas Vlachos and Isabelle Augenstein, 1139–1149. Dubrovnik, Croatia: Association for Computational Linguistics, May. <https://doi.org/10.18653/v1/2023.eacl-main.81>. <https://aclanthology.org/2023.eacl-main.81>.
- Kim, Minju, Beong-woo Kwak, Youngwook Kim, Hong-in Lee, Seung-won Hwang, and Jinyoung Yeo. 2022. *Dual Task Framework for Improving Persona-grounded Dialogue Dataset*. arXiv: 2202.05435 [cs.CL].
- LangChain. 2024. *LangChain Documentation*. <https://python.langchain.com/v0.2/docs/introduction/>.

- Lee, Young-Jun, Chae-Gyun Lim, Yunsu Choi, Ji-Hui Lm, and Ho-Jin Choi. 2022. “PERSONACHATGEN: Generating Personalized Dialogues using GPT-3.” In *Proceedings of the 1st Workshop on Customized Chat Grounding Persona and Knowledge*, edited by Heuseok Lim, Seungryong Kim, Yeonsoo Lee, Steve Lin, Paul Hongsuck Seo, Yumin Suh, Yoonna Jang, Jungwoo Lim, Yuna Hur, and Suhyune Son, 29–48. Gyeongju, Republic of Korea: Association for Computational Linguistics, October. <https://aclanthology.org/2022.ccgpk-1.4>.
- Li, Jiwei, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. “A Persona-Based Neural Conversation Model.” In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, 994–1003.
- Lieberman, Henry. 1995. “Letizia: An Agent That Assists Web Browsing.” In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, 924–929.
- Lin, Xi Victoria, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, et al. 2022. “Few-shot Learning with Multilingual Generative Language Models.” In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, edited by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, 9019–9052. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, December. <https://doi.org/10.18653/v1/2022.emnlp-main.616>. <https://aclanthology.org/2022.emnlp-main.616>.
- Liu, Qian, Yihong Chen, Bei Chen, Jian-Guang Lou, Zixuan Chen, Bin Zhou, and Dongmei Zhang. 2020. “You Impress Me: Dialogue Generation via Mutual Persona Perception.” In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, edited by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, 1417–1427. Online: Association for Computational Linguistics, July. <https://doi.org/10.18653/v1/2020.acl-main.131>. <https://aclanthology.org/2020.acl-main.131>.

- Liu, Xiao, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. “GPT understands, too.” *AI Open*, ISSN: 2666-6510. <https://doi.org/https://doi.org/10.1016/j.aiopen.2023.08.012>. <https://www.sciencedirect.com/science/article/pii/S2666651023000141>.
- Liu, Yifan, Wei Wei, Jiayi Liu, Xianling Mao, Rui Fang, and Danyang Chen. 2022. “Improving Personality Consistency in Conversation by Persona Extending.” In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 1350–1359. CIKM '22. Atlanta, GA, USA: Association for Computing Machinery. ISBN: 9781450392365. <https://doi.org/10.1145/3511808.3557359>. <https://doi.org/10.1145/3511808.3557359>.
- Madotto, Andrea, Zhaojiang Lin, Chien-Sheng Wu, and Pascale Fung. 2019. “Personalizing Dialogue Agents via Meta-Learning.” In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, edited by Anna Korhonen, David Traum, and Lluís Màrquez, 5454–5459. Florence, Italy: Association for Computational Linguistics, July. <https://doi.org/10.18653/v1/P19-1542>. <https://aclanthology.org/P19-1542>.
- Manning, Christopher D, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.
- Meng, Yu, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. *Generating Training Data with Language Models: Towards Zero-Shot Language Understanding*. arXiv: 2202.04538 [cs.CL].
- Meta. 2024. *Introducing Meta Llama 3: The most capable openly available LLM to date*. <https://ai.meta.com/blog/meta-llama-3/>.

- Miller, Alexander H., Will Feng, Adam Fisch, Jiasen Lu, Dhruv Batra, Antoine Bordes, Devi Parikh, and Jason Weston. 2018. *ParlAI: A Dialog Research Software Platform*. arXiv: 1705.06476 [cs.CL].
- Ni, Jinjie, Tom Young, Vlad Pandealea, Fuzhao Xue, Vinay Adiga, and Erik Cambria. 2021. “Recent Advances in Deep Learning Based Dialogue Systems: A Systematic Survey.” *CoRR* abs/2105.04387. arXiv: 2105.04387. <https://arxiv.org/abs/2105.04387>.
- Nie, Yixin, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. *Adversarial NLI: A New Benchmark for Natural Language Understanding*. arXiv: 1910.14599 [cs.CL].
- O’Neill, James, Polina Rozenshtein, Ryuichi Kiryo, Motoko Kubota, and Danushka Bollegala. 2021. “I Wish I Would Have Loved This One, But I Didn’t – A Multilingual Dataset for Counterfactual Detection in Product Reviews.” In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- OpenAI. 2024. *OpenAI Platform*. <https://platform.openai.com/>. Accessed: 2024-06-11.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, et al. 2024. *GPT-4 Technical Report*. arXiv: 2303.08774 [cs.CL].
- Ou, Jiao, Jinchao Zhang, Yang Feng, and Jie Zhou. 2022. “Counterfactual Data Augmentation via Perspective Transition for Open-Domain Dialogues.” In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, edited by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, 1635–1648. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, December. <https://doi.org/10.18653/v1/2022.emnlp-main.106>. <https://aclanthology.org/2022.emnlp-main.106>.

- Pawar, Urja. 2022. "Counterfactuals and Their Evaluation." <https://medium.com/@urjapawar/counterfactuals-and-their-evaluation-574ef58d34ac>.
- Princeton University. 2020. "Lecture Slides: Dialogue Systems Part 2." Princeton University. <https://www.cs.princeton.edu/courses/archive/spring20/cos598C/lectures/lec15-dialogue2.pdf>.
- Prompt Engineering Guide. 2024. "Basics of Prompting." Prompt Engineering Guide. <https://www.promptingguide.ai/introduction/basics>.
- Qiu, Minghui, Feng-Lin Li, Siyu Wang, Xing Gao, Yan Chen, Weipeng Zhao, Haiqing Chen, Jun Huang, and Wei Chu. 2017. "AliMe Chat: A Sequence to Sequence and Rerank based Chatbot Engine." In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, edited by Regina Barzilay and Min-Yen Kan, 498–503. Vancouver, Canada: Association for Computational Linguistics, July. <https://doi.org/10.18653/v1/P17-2079>. <https://aclanthology.org/P17-2079>.
- Rachel, Renaissance. 2022. "Large Language Models: An Overview." <https://renaissancerachel.com/large-language-models-overview/>.
- Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. *Improving Language Understanding by Generative Pre-Training*. V. 1. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.
- Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. "Language models are unsupervised multitask learners." *OpenAI blog* 1 (8): 9.

- Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.” *CoRR* abs/1910.10683. arXiv: 1910.10683. <http://arxiv.org/abs/1910.10683>.
- Ribeiro, Rui, Joao P. Carvalho, and Luísa Coheur. 2023. *PGTask: Introducing the Task of Profile Generation from Dialogues*. arXiv: 2304.06634 [cs.CL].
- Roese, Neal J. 1997. “Counterfactual thinking.” *Psychological Bulletin* 121 (1): 133–148.
- Serban, Iulian Vlad, Chinnadhurai Sankar, Mathieu Germain, Saizheng Zhang, Zhouhan Lin, Sandeep Subramanian, Taesup Kim, et al. 2017. “A Deep Reinforcement Learning Chatbot.” *CoRR* abs/1709.02349. arXiv: 1709.02349. <http://arxiv.org/abs/1709.02349>.
- Shea, Ryan, and Zhou Yu. 2023. “Building Persona Consistent Dialogue Agents with Offline Reinforcement Learning.” In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, edited by Houda Bouamor, Juan Pino, and Kalika Bali, 1778–1795. Singapore: Association for Computational Linguistics, December. <https://doi.org/10.18653/v1/2023.emnlp-main.110>. <https://aclanthology.org/2023.emnlp-main.110>.
- Son, Hye Chan, Joong Hyuk Lee, and Chang Won Park. 2014. “Recognizing Counterfactual Sentences with Fact Probing.” In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, 1222–1231.

- Song, Haoyu, Yan Wang, Kaiyan Zhang, Wei-Nan Zhang, and Ting Liu. 2021. “BoB: BERT Over BERT for Training Persona-based Dialogue Models from Limited Personalized Data.” In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, edited by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, 167–177. Online: Association for Computational Linguistics, August. <https://doi.org/10.18653/v1/2021.acl-long.14>. <https://aclanthology.org/2021.acl-long.14>.
- Song, Yiping, Rui Yan, Xiang Li, Dongyan Zhao, and Ming Zhang. 2016. “Two are Better than One: An Ensemble of Retrieval- and Generation-Based Dialog Systems.” *CoRR* abs/1610.07149. arXiv: 1610.07149. <http://arxiv.org/abs/1610.07149>.
- Touvron, Hugo, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, et al. 2023. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. arXiv: 2307.09288 [cs.CL].
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. “Attention is all you need.” In *Proceedings of Advances in Neural Information Processing Systems*, vol. 30.
- Wang, Fucheng, Yunfei Yin, Faliang Huang, and Kaigui Wu. 2023. “Please don’t answer out of context: Personalized Dialogue Generation Fusing Persona and Context.” In *2023 International Joint Conference on Neural Networks (IJCNN)*, 1–8. <https://doi.org/10.1109/IJCNN54540.2023.10191649>.
- Weizenbaum, Joseph. 1966. “ELIZA - A Computer Program For the Study of Natural Language Communication Between Man and Machine.” *Communications of the ACM* 9 (1): 36–45. <https://doi.org/10.1145/365153.365168>.

- Welleck, Sean, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2019. “Dialogue Natural Language Inference.” In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, edited by Anna Korhonen, David Traum, and Lluís Màrquez, 3731–3741. Florence, Italy: Association for Computational Linguistics, July. <https://doi.org/10.18653/v1/P19-1363>. <https://aclanthology.org/P19-1363>.
- Williams, Adina, Nikita Nangia, and Samuel Bowman. 2018. “A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference.” In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, edited by Marilyn Walker, Heng Ji, and Amanda Stent, 1112–1122. New Orleans, Louisiana: Association for Computational Linguistics, June. <https://doi.org/10.18653/v1/N18-1101>. <https://aclanthology.org/N18-1101>.
- Winograd, Terry. 1972. *Understanding Natural Language*. Academic Press.
- Wolf, Thomas, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. *Transfer-Transfo: A Transfer Learning Approach for Neural Network Based Conversational Agents*. arXiv: 1901.08149 [cs.CL].
- Wu, Chen Henry, Yinhe Zheng, Xiaoxi Mao, and Minlie Huang. 2021. “Transferable Persona-Grounded Dialogues via Grounded Minimal Edits.” In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, edited by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, 2368–2382. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, November. <https://doi.org/10.18653/v1/2021.emnlp-main.183>. <https://aclanthology.org/2021.emnlp-main.183>.

- Wu, Chien-Sheng, Steven Hoi, Richard Socher, and Caiming Xiong. 2020. *TOD-BERT: Pre-trained Natural Language Understanding for Task-Oriented Dialogue*. arXiv: 2004.06871 [cs.CL].
- Xu, Jing, Arthur Szlam, and Jason Weston. 2022. “Beyond Goldfish Memory: Long-Term Open-Domain Conversation.” In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, edited by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, 5180–5197. Dublin, Ireland: Association for Computational Linguistics, May. <https://doi.org/10.18653/v1/2022.acl-long.356>. <https://aclanthology.org/2022.acl-long.356>.
- Zhang, Shikib, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, and Bill Dolan. 2018. “Personalizing Dialogue Agents: I have a dog, do you have pets too?” In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2204–2213.
- Zheng, Yinhe, Guanyi Chen, Minlie Huang, Song Liu, and Xuan Zhu. 2019. “Personalized Dialogue Generation with Diversified Traits.” *CoRR* abs/1901.09672. arXiv: 1901.09672. <http://arxiv.org/abs/1901.09672>.
- Zhou, Junkai, Liang Pang, Huawei Shen, and Xueqi Cheng. 2023. “SimOAP: Improve Coherence and Consistency in Persona-based Dialogue Generation via Over-sampling and Post-evaluation,” 9945–9959. January. <https://doi.org/10.18653/v1/2023.acl-long.553>.
- Zhu, Qingfu, Lei Cui, Weinan Zhang, Furu Wei, Yining Chen, and Ting Liu. 2018. “Retrieval-Enhanced Adversarial Training for Neural Response Generation.” *CoRR* abs/1809.04276. arXiv: 1809.04276. <http://arxiv.org/abs/1809.04276>.

APPENDIX

A.1. Case Study

Table A.1 and Table A.2 include sample responses generated by trained models. In Table A.1, there is no sentence regarding personal information in the response produced by the LMEDR study. The answer of the model trained in LMEDR of the PersonaChat dataset, augmented with the GPT 3.5 method, includes a counterfactual sentence, and its answer is persona-consistent. Although the response from Llama 2 is related to the query, 'I love flowers' is not included in the personal information.

Table A.1 Example of Response Generation on the PersonaChat

Persona	I volunteer at the homeless shelter. My father played violin professionally. I am a piano teacher on the side. I play several instruments. I enjoy exploring new parts of the city.
Query	It is a plant that closes when you touch it. I like a good violin.
Gold Response	Oh! Sorta like a venus fly trap? Those are neat.
Original	That's interesting! Do you have any pets?
GPT 3.5	That is cool! I wish I could play the violin like my father.
Llama 2 13B	That is cool ! I love flowers too!

In Table A.2, here is another example showing that he is a college student according to the original study's findings. Still, such information is not included in his personality traits. A persona-consistent response was produced in models trained in LMEDR of the PersonaChat dataset, augmented with GPT 3.5 and Llama separately.

Table A.2 Example of Response Generation on the PersonaChat

Persona	I have an internet addiction and spend a lot of time playing role-playing games. I faint at the sight of blood. My dream in life is to work from home. I went to school to be a veterinarian but hated it.
Query	So tell me more about yourself.
Gold Response	I play video games . Would love to make a living out of it.
Original	I am a college student.
GPT 3.5	I like roleplaying games. I wish I could work from home.
Llama 2 13B	I like to role play .