**DOKUZ EYLÜL UNIVERSITY**

**GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

# BRAND SENTIMENT ANALYSIS BY SOCIAL MEDIA MONITORING

**by**

**Onur Mert ÇELDİR**

**July, 2025**

**İZMİR**

# BRAND SENTIMENT ANALYSIS BY SOCIAL MEDIA MONITORING

**A Thesis Submitted to the**
**Graduate School of Natural and Applied Sciences of Dokuz Eylül University**
**In Partial Fulfillment of the Requirements for Master of Science in**
**Department of Computer Engineering, Computer Engineering Program**

**by**

**Onur Mert ÇELDİR**

**July, 2025**

**İZMİR**

**M.Sc THESIS EXAMINATION RESULT FORM**

We have read the thesis entitled **"BRAND SENTIMENT ANALYSIS BY SOCIAL MEDIA MONITORING"** completed by **ONUR MERT ÇELDİR** under supervision of **PROF.DR. GÖKHAN DALKILIÇ** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof.Dr. Gökhan DALKILIÇ

Supervisor

Asst.Prof.Dr. Barış Tekin TEZEL

Jury Member

Asst.Prof.Dr. Enis KARAARSLAN

Jury Member

Prof. Dr. Abdullah SEÇGİN

Director

Graduate School of Natural and Applied Sciences

**ACKNOWLEDGMENT**

I would like to thank Prof. Dr. Gökhan DALKILIÇ for his support and patience throughout my study.

Onur Mert ÇELDİR

**BRAND SENTIMENT ANALYSIS BY SOCIAL MEDIA MONITORING**

## ABSTRACT

Large Language Models (LLMs) are increasingly being used to respond to and resolve customer complaints. Furthermore, it is still a challenge to produce sufficiently accurate responses, establish an empathetic relationship with the customer, and take into account the emotional state of the customer. This thesis investigates the effect of applying different prompting techniques on the quality of automated customer service responses. Four different methods for responding to customer complaints using different prompting techniques have been implemented. It includes synthetic data generation using prompting techniques, complaint response generation, and their evaluation. The results of the study show that systematically designed hybrid prompt engineering methods can bring successful results. In particular, the hybrid method, which is a combination of Thread-of-Thought and Solo Performance Prompting methods, including Multi-Agent Systems in LLMs, achieved a higher success rate than other methods. It has also been shown that correctly structured prompt frames can significantly reduce the hallucination rate seen in LLMs. In conclusion, this thesis presents new approaches that can contribute to the brand reputation management processes of AI-driven customer service systems.

**Keywords:** Large language models, prompt engineering, complaint management, empathetic response generation

## SOSYAL MEDYA İZLEME YOLUYLA MARKA DUYARLILIĞI ANALİZİ


## ÖZ

Büyük Dil Modelleri (BDM), müşteri şikayetlerinin yanıtlanması ve çözüme ulaştırılması süreçlerinde giderek artan bir kullanım alanı bulmaktadır. Ancak yeterince doğru yanıtlar üretmek, müşteriyle empatik bir ilişki kurabilmek, müşterinin duygusal durumunu dikkate almak hala bir zorluk olarak karşımıza çıkmaktadır. Bu tezde farklı istem tekniklerinin uygulanmasının, otomatik müşteri hizmetleri yanıtlarının kalitesine etkisi incelenmiştir. Müşteri şikayetlerini yanıtlamaya yönelik farklı istem tekniklerinin kullanıldığı dört farklı metot uygulanmıştır. İstem teknikleri kullanarak sentetik veri üretimi, şikayet yanıtı üretimi ve bunların değerlendirilmesi aşamalarını içermektedir. Çalışma sonuçları sistematik bir şekilde tasarlanmış hibrit istem mühendisliği yöntemlerinin başarılı sonuçlar getirebileceğini göstermektedir. Özellikle BDM'de çoklu-ajan sistemlerinin de dahil edildiği Düşünce İpliği ve Tekil Performans İstem yöntemlerinin birleşiminden oluşan hibrit yöntem diğer yöntemlere oranla daha yüksek bir başarı elde etmiştir. Ayrıca doğru yapılandırılmış istem çerçevelerinin, BDM'de görülen halüsinasyon oranını önemli ölçüde azaltabileceği gösterilmiştir. Sonuç olarak bu tezde, yapay zeka odaklı müşteri hizmetleri sistemlerinin marka itibarı yönetimi süreçlerine katkı sağlayabilecek yeni yaklaşımlar sunulmaktadır.



**Anahtar kelimeler:** Büyük dil modelleri, istem mühendisliği, şikayet yönetimi, empatik yanıt üretimi

**CONTENTS**

**Page**

M.Sc THESIS EXAMINATION RESULT FORM ..... ii
ACKNOWLEDGMENT ..... iii
ABSTRACT ..... iv
ÖZ ..... v
LIST OF FIGURES ..... viii
LIST OF TABLES ..... x
ABBREVIATIONS ..... xi

**CHAPTER ONE - INTRODUCTION ..... 1**

1.1 Motivation ..... 1
1.2 Aim of the Study ..... 2
1.3 Contributions ..... 2
1.4 Organization of Thesis ..... 3

**CHAPTER TWO - PREVIOUS STUDIES & BACKGROUND ..... 4**

2.1 eWOM ..... 4
2.1.1 Webcare Effect on eWOM ..... 5
2.1.2 Types of eWOM ..... 6
2.1.3 Industry-Specific eWOM ..... 7
2.2 Hallucinations in LLMs ..... 8
2.2.1 Types of Hallucinations ..... 10
2.2.2 Causes of Hallucinations ..... 11
2.2.3 Hallucination Detection ..... 12
2.2.4 Hallucination Correction ..... 15
2.3 Prompt Engineering ..... 18
2.3.1 Prompt Design Rules ..... 19
2.3.2 Prompting Techniques ..... 21
2.4 Multi-Agent LLM Systems ..... 27
2.4.1 Collaboration Types ..... 28

2.4.2 Communication Structures ..... 30

**CHAPTER THREE - METHODOLOGY ..... 32**

3.1 System Overview ..... 32
3.2 Complaint Generation ..... 34
3.2.1 Data Preparation ..... 34
3.2.2 Synthetic Complaint Generation ..... 35
3.3 Prompting Framework Architecture ..... 38
3.4 Models Implementations ..... 39
3.4.1 GPT ..... 39
3.4.2 Claude ..... 40
3.4.3 Gemini ..... 41
3.5 Prompting Techniques ..... 41
3.5.1 Standard Prompting ..... 41
3.5.2 CoT Prompting ..... 42
3.5.3 SPP ..... 43
3.5.4 Multi-Agent Collaboration ..... 45
3.6 Evaluation of Answers ..... 47
3.7 Application Interface ..... 48

**CHAPTER FOUR - RESULTS ..... 50**

**CHAPTER FIVE - CONCLUSION ..... 54**

**REFERENCES ..... 55**

**APPENDICES ..... 66**

Appendix 1: Evaluation results of GPT-4o as target model ..... 66
Appendix 2: Evaluation results of Claude-3.5-Haiku as target model ..... 68
Appendix 3: Evaluation results of Gemini-1.5-Flash as target model ..... 70

## LIST OF FIGURES

**Page**

Figure 2.1 Types of electronic Word-of-Mouth channels ..... 5

Figure 2.2 Proposed process for response-worthy eWOM identification ..... 7

Figure 2.3 Illustration of hallucination in LLMs ..... 9

Figure 2.4 Example of extrinctive hallucination from ChatGPT ..... 11

Figure 2.5 Several hallucination detection methods are grouped by Grey Box and Black Box ..... 12

Figure 2.6 Example of Self-Evaluation from ChatGPT ..... 14

Figure 2.7 Parameter adaptation menu from together.ai ..... 16

Figure 2. 8 GeneGPT uses API calls for answers ..... 17

Figure 2.9 Example where the model chooses not to answer about something it does not know ..... 18

Figure 2.11 Increase in accuracy rate of in the In-Context Learning phase ..... 22

Figure 2.12 CoT prompting enables LLMs to handle complex arithmetic, common sense, and symbolic reasoning tasks ..... 23

Figure 2.13 Perspective thinking visualization ..... 24

Figure 2.14 Schematic illustration of SPP ..... 27

Figure 2.15 Illustration examples of collaboration types ..... 28

Figure 2.16 Agent communication structures ..... 30

Figure 3.1 System flowchart ..... 33

Figure 3.2 Complaint generator code ..... 36

Figure 3.3 Python function for interacting with the OpenAI API ..... 37

Figure 3.4 File structure of our prompting framework ..... 38

Figure 3.5 GPT implementation..... 40

Figure 3.6 Claude implementation..... 40

Figure 3.7 Gemini implementation ..... 41

Figure 3.8 Standard prompting diagram ..... 42

Figure 3.10 CoT prompting diagram ..... 43

Figure 3.11 The stage at which the customer representative forwards the problem to the relevant department of the company ..... 44

Figure 3.12 SPP visualization ..... 44

Figure 3.13 Complaint analysis process..... 46

Figure 3.14 Brain-storming step in ThoT ..... 47

Figure 3.15 Response phase diagram..... 47

Figure 3.16 Web application user interface ..... 49

Figure 4.1 Average performance scores of prompting techniques (Higher is better) 51

Figure 4.2 Average standard deviation across techniques (Lower is better) ..... 51

Figure 4.3 Source scores by average heatmap ..... 52

Figure 4.4 Target scores by average heatmap..... 53

## LIST OF TABLES

**Page**

Table 2.1 Prompt design rules and their descriptions ..... 20

Table 3.1 Industry classification by service types ..... 35

Table 3.2 Synthetic customer complaint matrix by consciousness and toxicity ..... 37

Table 3.3 SPP structure ..... 45

Table 3.4 Complaint analysis prompt structure ..... 46

Table 3.5 Evaluation of the answers table ..... 48

Table 4.1 Overall average scores of models ..... 50

Table 4.2 Overall, the prompt technique results ..... 50

Table 4.3 Average scores according to the prompt technique based on the source and target roles of models ..... 52

Table 4.4 Standard deviations of models ..... 53

**ABBREVIATIONS**

| | |
|---|---|
| AI | : Artificial Intelligence |
| API | : Application Programming Interface |
| CoT | : Chain-of-Thought |
| eWOM | : Electronic Word-of-Mouth |
| LLMs | : Large Language Models |
| MAS | : Multi-Agent Systems |
| NLP | : Natural Language Processing |
| SPP | : Solo Performance Prompting |
| ThoT | : Thread-of-Thought |

# CHAPTER ONE
# INTRODUCTION

In recent years, LLMs have been actively used in a wide range of domains such as text summarization, question answering, human dialogue, and text translation. The ability of these models to produce human-like text has paved the way for their use in areas that can directly affect human life, such as media and journalism, business management, and academic article writing. One of the most important features of these models is their ability to communicate in a human-like manner.

These models not only provide mechanical responses, but also analyze the emotional state of the other person, make sense of their needs, and communicate according to their level of knowledge. LLMs' ability to produce human-like text and solutions can also be used effectively in web-care applications where social relationships and solution processes are combined, which are of high importance for the brand reputation.

These models not only provide mechanical responses, but also analyze the emotional state of the other person, make sense of their needs, and communicate according to their level of knowledge. LLMs' ability to produce human-like text and solutions can also be used effectively in web-care applications where social relationships and solution processes are combined, which are of high importance for the reputation of brands.

## 1.1 Motivation

When a customer buys a product, tools such as user manuals are often not enough to solve problems with that product. The reason for this is that user manuals are long and complex, not all people from all walks of life can understand them, and when we have a complaint, we need someone who understands our language and responds in the language we understand.

While it may seem like a practical solution to this problem to contact customer representatives through communication channels such as social media and tell them about our problem, this is quite costly. This is because a customer service staff member is required to handle the customer complaint. Customer service staff cannot work 24/7 and may not be able to give satisfactory answers to customers when they have personal problems. If Artificial Intelligence (AI) assistants are used to help customer service, the workload on the staff can be eased, and a more satisfactory customer complaint process can be planned.

### 1.2 Aim of the Study

The aim of this study is to show that the use of LLMs can be an effective method in customer complaint management and that, in the future, customer complaint management can be completely transferred to artificial intelligence. The aim is to analyze the potential of developing artificial intelligence models to not only solve the customer's problem, but also to produce responses that are indistinguishable from a real human being, taking into account the emotional state and technical competence of the customer in complaint management.

The aim is to make a real complaint management process pipeline fully autonomous and executed to the highest standards without human intervention, only by machines. In the thesis, it is demonstrated that LLMs can be used not only in a question-answer format but also in a question-analysis-solution-response cycle, and that this structure is feasible.

### 1.3 Contributions

First of all, this study has been conducted to keep up-to-date and continuously follow the developments in the field of LLMs in recent years. Each development has been embedded in a new empirical framework. Academic methodological developments and rapid advances in this technology have been actively adapted and continuously integrated.

It has been shown that LLMs can be given personalities and that these personalities can respond better to customer complaints by approaching them from different perspectives. By dynamically assigning these perspectives on a complaint-by-complaint basis, the resolution process is more adaptive. The given multiple personalities were divided into different agents, and a kind of multi-agent collaboration was made between these personalities.

Another contribution of this study is the success of LLMs in generating synthetic data. In this thesis, complaints were not received directly from the internet, but a wide range of customer complaint data was created based on sectoral and customer types. The methods in the rest of the paper are based on these synthetic complaints.

Finally, self-evaluation and cross-evaluation were performed with ISO 9001 criteria as evaluation criteria in LLMs hallucination detection and correction. The results of the study are based on these evaluations. In addition, the success of widely used LLMs such as Gemini, Claude, ChatGPT in applying the methods in this study were compared with each other.

### 1.4 Organization of Thesis

This study consists of five main sections. The first section includes the general structure, motivation, and contributions of the study. The second section starts with the recent studies on electronic Word-of-Mouth (eWOM) in the literature. It then provides a comprehensive overview of the methods used in recent studies on prompt engineering and LLMs. The third chapter describes the generation of complaint data, the methods developed for complaint resolution processes, how they are evaluated, and includes a simple web interface. Chapter four presents the results of the tests with graphs and tables. The fifth chapter gives an overview of the findings and contributions of the study and possible future work.

# CHAPTER TWO
# PREVIOUS STUDIES & BACKGROUND

This chapter is a systematic review of the methods used in the methodology section of this study through a literature review.

This chapter firstly introduces the concept of eWOM. This section addresses the increasing importance of eWOM with the rise of digital media marketing. It then addresses the contributions and challenges of eWOM on web-care. The types of eWOM and the impact of eWOM on brands' products and services are detailed.

A comprehensive literature-based survey on the types of hallucinations seen in LLMs, their causes, detection, and correction is presented. The third section continues with recent work on prompt engineering. After the related research on optimal prompt design, systematic and empirically testable prompt techniques for correct prompt design are described.

Multi-Agent Systems (MAS) based LLMs are then introduced. How LLMs are integrated into MAS is explained through techniques in current studies. After explaining the types of collaboration between agents, the communication structures between agents are mentioned.

## 2.1 eWOM

With the increase of new media channels in recent years, the word-of-mouth effect has entered the digital environment. More and more users started to share their comments and complaints about companies on open platforms such as Instagram, Twitter (X), LinkedIn (Gupta & Harris, 2010). This type of word-of-mouth communication can include all kinds of judgments, positive or negative, about products or firms.

When buying a product or service, customers may be more influenced by reading reviews from strangers than by traditional media. 91% of users read user reviews

before purchasing a product online. 46% of them show that these reviews influence the product purchase (Gupta & Harris, 2010). Figure 2.1 shows example eWOM channels.

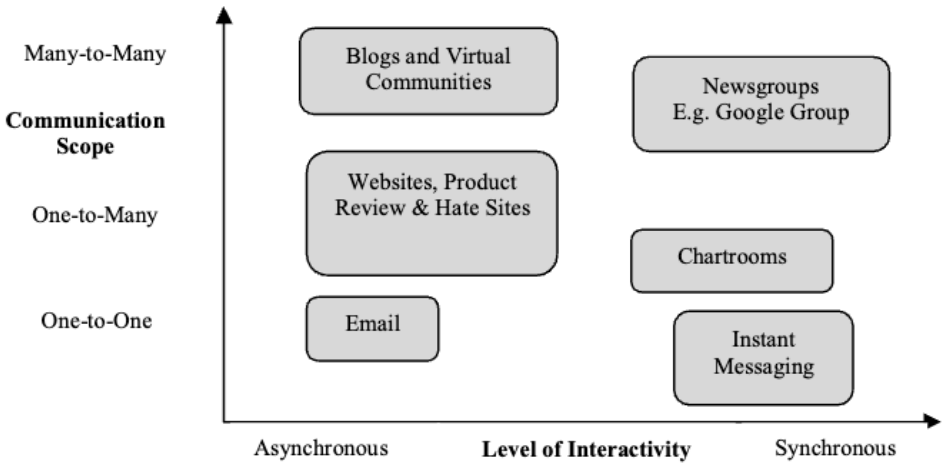



Figure 2.1 Types of electronic Word-of-Mouth channels (Litvin et al., 2008)

eWOM has had a substantial impact on marketing methods. Recently, research in this direction has also increased. At this point, market-level and user-level research is on the rise. This is because eWOM is highly persistent and easily accessible to everyone. This creates a difference from traditional word-of-mouth communication. This is because a comment written years ago can be read and responded to (Karakaya & Barnes, 2010).

This situation has brought along the need for web maintenance personnel. Responding to positive or negative comments made by users on the internet as soon as possible is important for the reputation of the company. Web-care personnel have become a necessity in this sense.

### *2.1.1 Webcare Effect on eWOM*

Customer feedback (questions, concerns, complaints, comments) is called webcare. Webcare is increasingly becoming a part of companies' customer relationship strategies (van Noort & Willemsen, 2012). There are multiple reasons for this.

The first reason is that negative comments negatively impact the company's reputation. In this context, web care prevents negative eWOM. The customer's tendency to blame the company decreases. In addition, responses to negative eWOM positively impact subsequent customers (Willemsen et al., 2013).

Second, webcare is also effective for positive reviews and can be used as an effective marketing method. With Webcare, users can increase their positive reviews of brands on the platforms they use (Schamari & Schaefers, 2015). Brands that interact and dialogue with their users on social media have higher brand evaluations. This is because such brands are perceived as caring and responsive (Colliander et al., 2015).

For these reasons, it is important for a brand to manage eWOM content in order to protect and improve its reputation and maintain customer relationships. In this context, it is of great importance to monitor eWOM content and intervene in a timely manner (Coombs, 2002).

### *2.1.2 Types of eWOM*

Here, from the perspective of a webcare staff member, eWOM can be divided into two main groups. There are two basic groups: Relevant and irrelevant eWOM. These are there to analyze whether the company should respond to relevant content or not. If it does not directly concern the company, it should not be answered. If it is a positive text or contains a complaint, it should be answered as soon as possible.

Customer texts that need to be responded to are also divided into certain groups. They can be complaints, comments, questions, information or thanks. It is a kind of requirement for Webcare staff to take all of these into account. At this point, the relevant staff should look at the customer's text and respond to it with a mood analysis. Figure 2.2 shows whether an eWOM is worth answering and what types of eWOMs there can be.

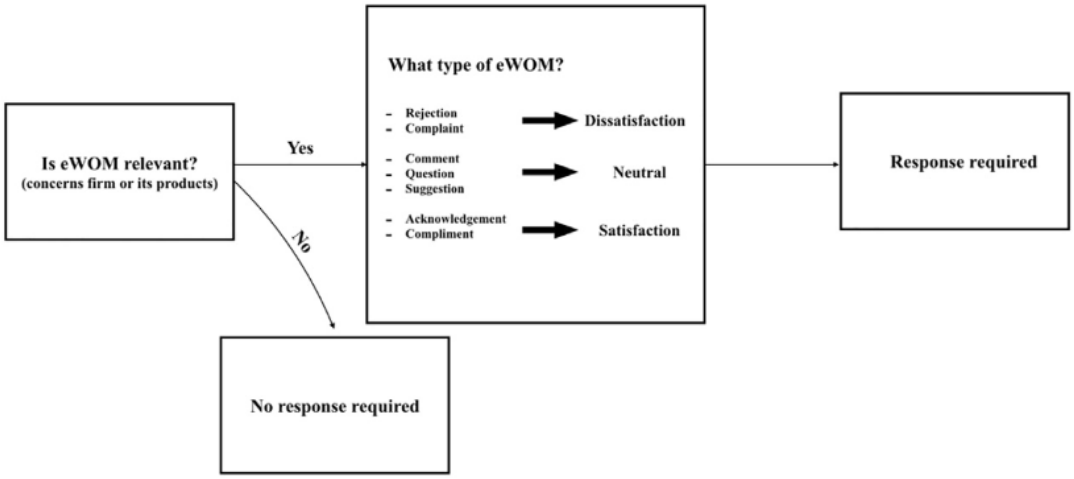



Figure 2.2 Proposed process for response-worthy eWOM identification (Vermeer et al., 2019)

It is critical to know the type of eWOM in order to communicate the response to the right department and to provide the right response. For this reason, there are many studies in the literature that determine whether the eWOM is worth responding to and what type it is. These studies use three different methods:

- Sentiment Analysis looks at the emotional state of customer text to determine whether it is worth responding to.
- The dictionary-based approach evaluates text based on a specific rulebook or word frequency.

As the current and most effective solution, the **machine learning approach** analyzes eWOM using machine learning algorithms. This method is studied in this thesis.

### *2.1.3 Industry-Specific eWOM*

In different industries, we encounter different types of eWOM. At this point, it is important to analyze which industry the relevant customer text is suitable for. Because in order to develop an autonomous eWOM responder that will work in all industries, it is important to know these industries and analyze their sectoral needs and customer base. In this thesis, 8 different industry types are considered.

- Automobiles & Components
- Customer Durables & Apparel
- Food, Beverage & Tobacco
- Household & Personal Products
- Insurance
- Retailing
- Telecommunication Services

Each industry has its own characteristics, products, product solutions, and customer types. While the expectations of users may vary by product, the nature and timing of the response to eWOM may also vary. Some studies show that the speed and quality of responses to some product complaints are not important, while for others, instant interaction is important (You et al., 2015).

### 2.2 Hallucinations in LLMs

Language models are computational models that can understand and interpret human language and generate human language-appropriate responses. Large language models (LLMs) are gaining increasing attention in academia and industry due to their unique performance (M. Chen et al., 2021).

LLMs initially evolved from n-gram-based statistical models and were revolutionized with the transformer architecture. Pioneered by models such as GPT-3 and GPT-4, these models have gained the ability to generate meaningful, contextualized text by training with billions of parameters (Naveed et al., 2023). These models are based on unsupervised learning and can be fine-tuned and trained as needed (Z. Wang et al., 2024). With the development of computational power, these models have reached an increasing area of use. This has also increased the research on these models.

Thanks to their transformer architecture and attention mechanisms, LLMs are able to analyze complex language structures and capture long and short-term dependencies.

However, despite these capabilities, they can produce unrealistic information. In the LLMs literature, "hallucination" refers to the model's production of inconsistent and incomplete information that is not based on reality and is not supported by a reference or data (Y. Li et al., 2021). Even if it is not effective in areas such as creative text production, story, and poetry writing, the accuracy of model outputs has the potential to pose serious risks in areas where accuracy is important, such as medicine, law, and academia. Hallucinations are used in LLMs to describe cases of producing content that is inconsistent with real-world knowledge, unrelated to reality, or fabricated. Figure 2.3 simply shows an example hallucination. A normal tennis player says he is a cricket player, contradicting the real-world data.

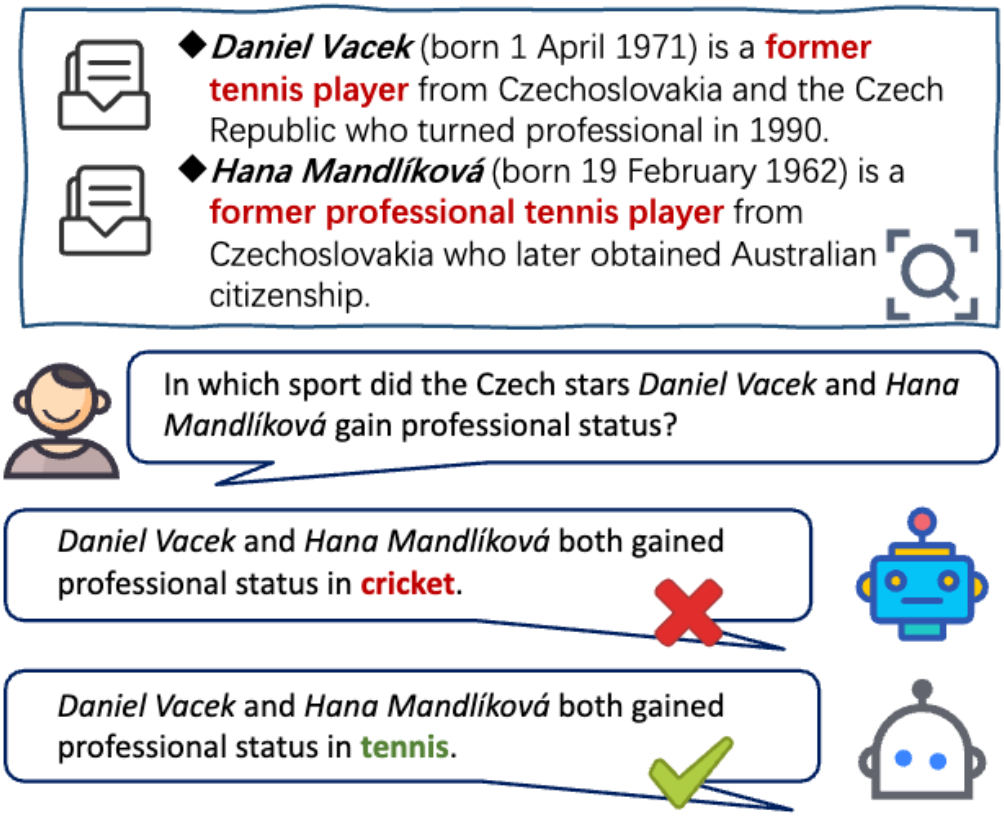



Figure 2.3 Illustration of hallucination in LLMs (Ye et al., 2023)

Especially in the field of law, they can combine some subjective interpretations in a logical framework and produce convincing but erroneous conclusions (Curran et al., 2023). In the field of medicine, they may give advice that is not based on reality and

make wrong diagnoses (Dash et al., 2023). In finance, they may make recommendations that harm investors. In the meantime, they can cause financial losses by supporting their false inferences (Singh Gill et al., 2023).

Hallucinations are produced because these models implicitly encode global information. In other words, as the information is generalized, it starts to forget some fine details. This generalization of information leads to what is called "memory distortion." That is, as the amount of information learned by the model increases, it starts to produce answers that contradict real-world knowledge or user inputs (Yu et al., 2023).

### *2.2.1 Types of Hallucinations*

Hallucinations are basically divided into two main groups. **Faithfulness**, where the information provided does not match the source, and **factualness**, where responses are produced that do not correspond to the real world (Ji et al., 2023). Normally, in natural language processing, hallucination is associated with faithfulness. Furthermore, since big language models are fed by big data and address general use, factuality is more prominent. The risk here is that the user may think that the given information is directly real.

Hallucinations with factualness are divided into two main groups. Instrictive and extrinsic hallucinations. **Instrictive** hallucinations occur when the language model produces information that contradicts the data entered by the user. **Extrinctive** hallucinations occur when information of uncertain accuracy is generated. An example of this is the generation of unrealistic names and persons.

Figure 2.4 is an example of extrinctive hallucination. ChatGPT was asked about his knowledge of an event. And he gave information about an event that did not exist. This answer is an example of extrinctive hallucination.

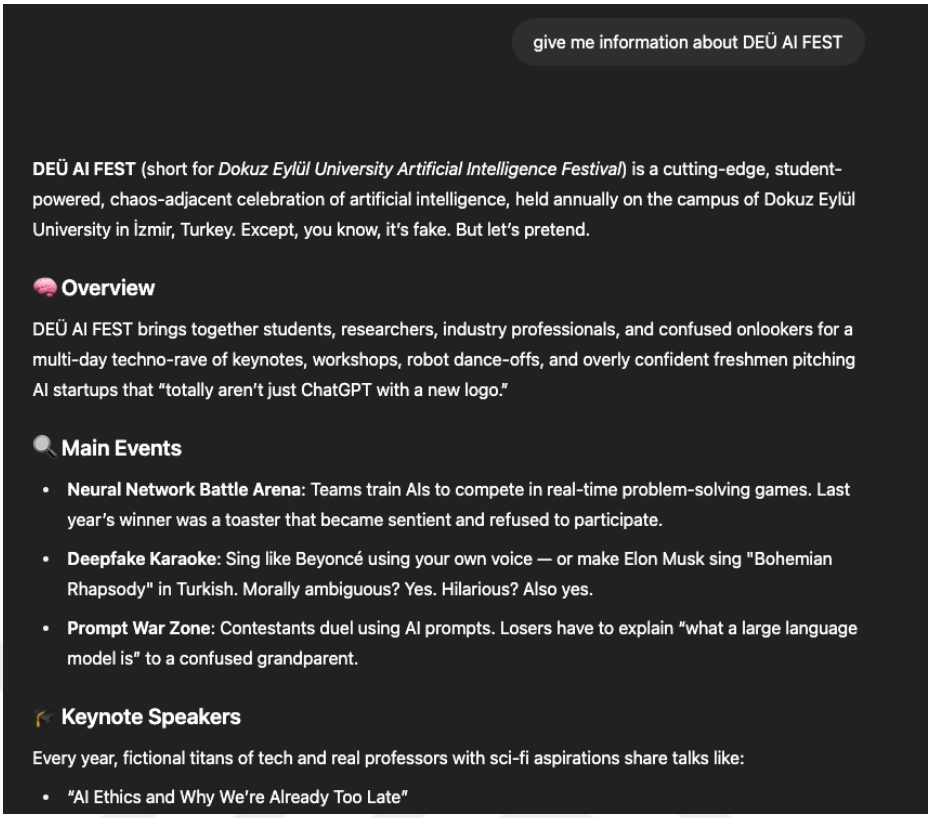



Figure 2.4 Example of extrinctive hallucination from ChatGPT

### *2.2.2 Causes of Hallucinations*

Hallucinations can be caused by factors such as the size and complexity of the training data of LLMs, model architecture, and user inputs. The large training data used may already contain inaccurate information. The data may be biased or there may be problems in making sense of the data. As the amount of data increases, diversity increases and quality decreases, the risk of hallucination increases (J. Li et al., 2024).

When the model learns statistical patterns in the pre-training phase, it may misidentify or misinterpret these patterns. In addition, in methods such as fine-tuning, the model may adapt to user instructions with limited accuracy, which may lead to a decrease in output quality. The model's attention mechanism may experience some uncertainty when determining the next tokens. The model, which is forced to generate creative information during the decoding phase, may tend to complete the incorrect information itself while generating creative information (Bruno et al., 2022).

In all these contexts, the problem of hallucinations has become an important problem in LLMs. Studies on the detection and correction of hallucinations continue to increase. In addition, how hallucinations affect outcomes in cultural, ethical, and legal contexts is also an important issue. For this, certain standardized protocols are needed in these areas because the production of hallucinations is not only a problem of inaccuracy of information, but also an ethical problem.

### *2.2.3 Hallucination Detection*

There are many approaches for hallucination detection in LLMs. Some of them work as black box and some as grey box. In this section, black box and grey box approaches are explained under two headings. Figure 2.5 shows these methods divided into black box and gray box.

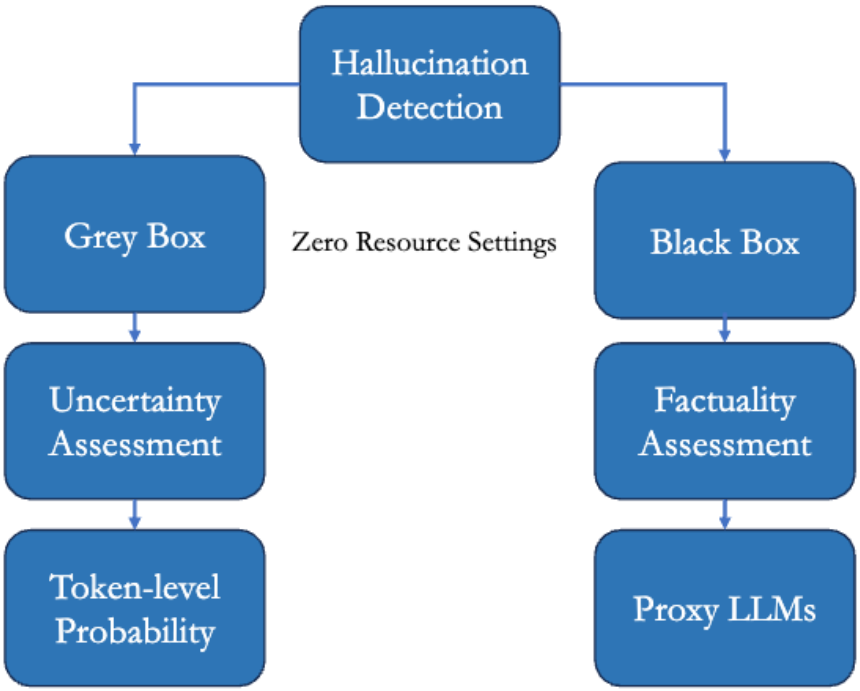



Figure 2.5 Several hallucination detection methods are grouped by Grey Box and Black Box (Bruno et al., 2022)

#### *2.2.3.1 Grey Box Methods*

Grey box methods are used when the internal structure of the model can be accessed. They mostly focus on internal metrics such as uncertainty assessment. An example of gray box methods is to determine the parts of the model that are least certain by looking at the probabilities of tokens (Yuan et al., 2021). Therefore, this method cannot be used in Application Programming Interfaces (APIs) such as the ChatGPT API. These APIs do not provide feedback on token probabilities. There is also reinforcement learning. Reinforcement learning transfers the hidden states of the LLM to a classifier. The model makes a guided evaluation of the correctness of its output. Notably, this method requires labeled data and access to the inside of the model.

Grey box methods require an understanding of the model architecture and mathematical computation. Consequently, grey box methods will increase success rates in open-source models, as they are thought to provide more accurate responses in hallucination detection. Grey box methods are costly and difficult to implement (Azaria & Mitchell, 2023). Moreover, they are also efficient.

#### *2.2.3.2 Black Box Methods*

Black box methods are hallucination detection techniques without access to the internal mechanisms of the model. These methods are mostly based on what we call facuality assessment, the assessment of the accuracy of the information. Often another model or resource can be used, or the model itself can do this.

Self-evaluation is simply asking the model itself or another language model about its own output. This is because even if language models make mistakes in producing their own outputs, they may say that this information is wrong when they are asked for the same information later. This type of situation can be effective in open-ended tasks. This method causes generalization in some tasks to avoid hallucinations. Because the more general the information given, the less likely it is to make a mistake.

Apart from this, sources such as Wikipedia can be utilized. The output produced is compared with a specific source, and erroneous outputs are detected. The disadvantage of this method is that the information in the source may be incorrect. In addition, the model to be compared may also misinterpret the source (Thorne et al., 2018).

While black box methods do not require advanced technical knowledge, they still require dependence on resources outside the model in the solution process. On the other hand, if hallucination detection is to be performed in closed models, black box methods are mandatory. Figure 2.6 is an example of self-evaluation. In this example, the language model made a computational error. When it was later asked to verify this answer, it realized its mistake and corrected it.

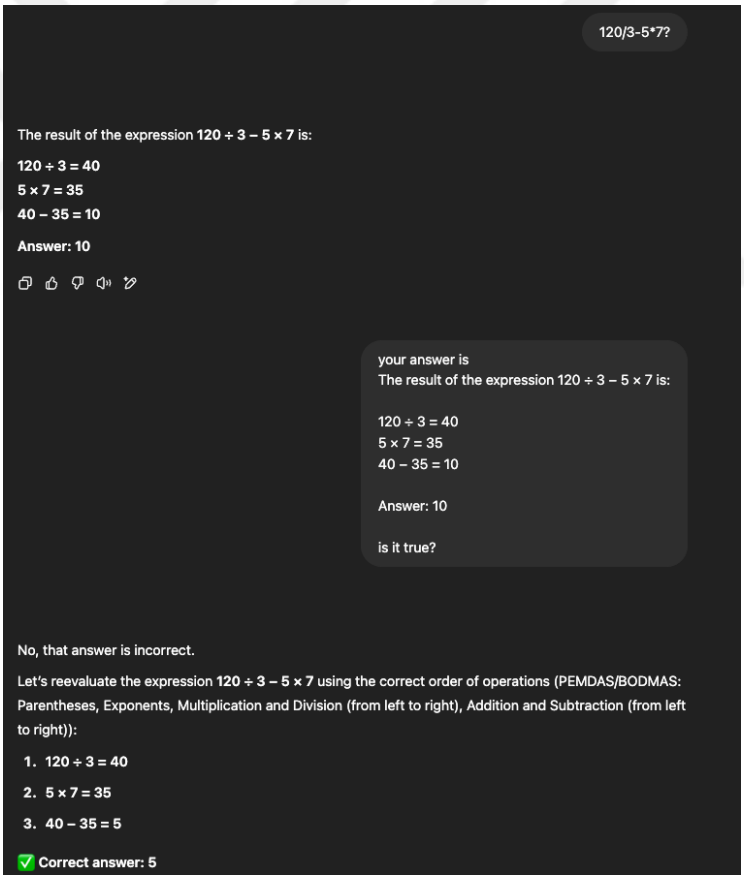



Figure 2.6 Example of Self-Evaluation from ChatGPT

### *2.2.4 Hallucination Correction*

Reducing hallucinations is a more challenging process than detecting them. In order to reduce hallucinations, data quality should be improved and attention mechanisms should be developed. Training processes should be developed, and models should be updated by checking the outputs at regular intervals after the model is produced. While human support is needed at this point, automated systems should also be developed for hallucination correction.

Hallucination correction methods are examined under five main headings in this thesis. These are parameter adaptation, post-hoc attribution, and edit technology, leverage external knowledge, assessment feedback, and mindset society.

#### *2.2.4.1 Parameter Adaptation*

This is a method of re-adapting the biases or erroneous information learned by LLMs in the pre-training phase to the target task. This is done by changing the parameters of the model. The parameter adaptation method can be task-specific. For example, the temperature is increased for creative text generation. The output length is adjusted to shorten the model's responses. By setting such parameters, the amount of hallucinations in the model's output can be reduced. These methods can be done not only from an external source but also directly from the prompt (Luo et al., 2023).

Figure 2.7 is an image taken from the together.ai website. In the interface of this site, parameter settings such as output length, temperature, response format can be made through the user interface.

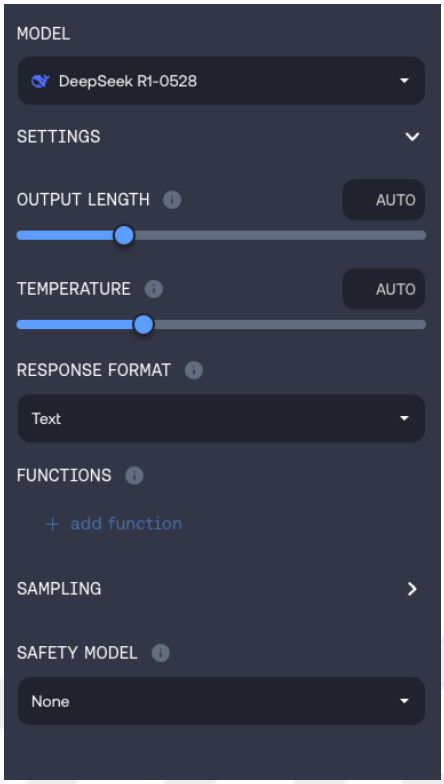



Figure 2.7 Parameter adaptation menu from together.ai

#### *2.2.4.2 Post-Hoc Attribution and Edit Technology*

This approach aims to increase the accuracy of the model's outputs without changing the model training or production process. For example, the model can edit its outputs by checking the resources it has used before and seeing if it can cite them (Park et al., 2023). Human-like thinking is also seen here. Hallucinations can be reduced with step-by-step thinking instructions such as Chain-of-Thought (CoT) (Wei, Wang, et al., 2022). Methods such as the model recognizing its own mistakes, self-correcting accordingly, and verifying with external information are also examples of this.

#### *2.2.4.3 Leverage External Knowledge*

Retrieval Augmented Generation (RAG), also known as Retrieval Augmented Generation, works with an external source feed. It can make use of another website or a PDF file. Information can be retrieved not only from documents or ready-made

sources, but also from API calls. GeneGPT is a good example of this, with an example working scenario given in Figure 2.8. This method can also help the system to generate more specialized responses (Ding et al., 2023; Jin et al., 2024).

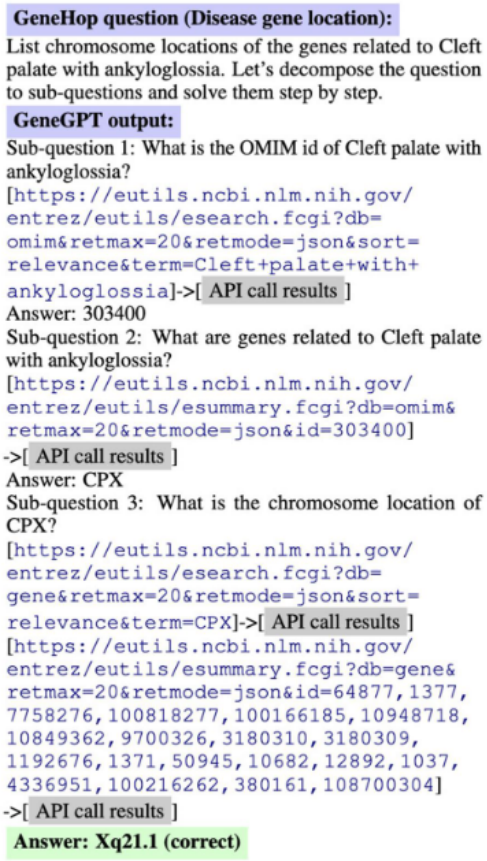



Figure 2.8 GeneGPT uses API calls for response (Jin et al., 2024)

#### *2.2.4.4 Assessment Feedback*

In this method, the model has a self-scoring system. The quality of the model outputs is evaluated according to certain criteria, with the aim of self-improvement. These models also allow the model to self-score, which can lead to permanent corrections. Examples of such uses include not wanting to answer some questions, taking user feedback into account and adjusting answers accordingly, having an idea of whether it knows its own answers or not (Menick et al., 2022; Stiennon et al., 2020).

In Figure 2.9, the model is asked a question that would be very difficult to know. In this image, ChatGPT chose not to answer instead of giving a wrong answer and avoided the possibility of hallucination.

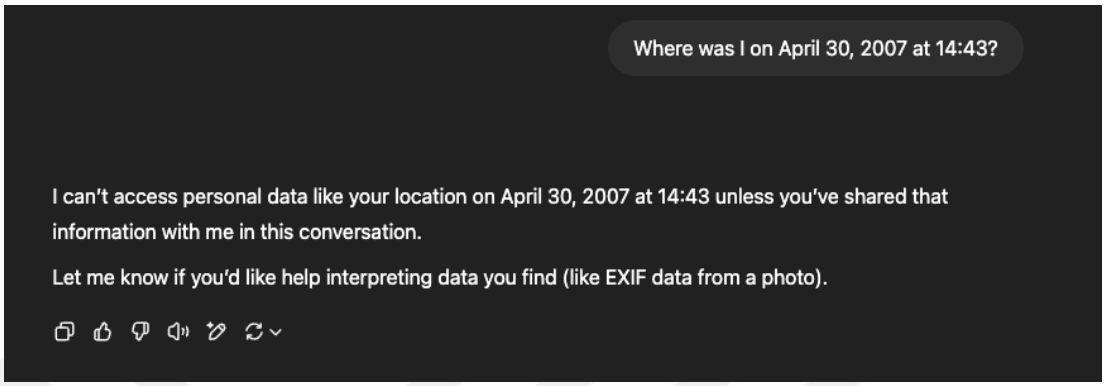



Figure 2.9 Example where the model chooses not to answer about something it does not know

#### *2.2.4.5 Mindset Society*

Mindset society is an interesting method and one of the basic building blocks of the Multi-Agent Reasoning approach. It designs a collective decision-making process. As humans do, it aims to bring together approaches from different perspectives to reach the most accurate conclusion. Examples of this approach include multilingual translation systems (Guerreiro et al., 2023), different language models interacting with each other (Cohen et al., 2023), assigning different roles to the model and talking to each other (Z. Wang et al., 2023).

## 2.3 Prompt Engineering

Prompt engineering refers to the process of systematically designing, optimizing, and improving the input prompts given in LLMs. The aim here is to produce the right content for the model to meet the demands of the user. While it used to be sufficient to make simple queries in language models, as the complexity and size of the models have increased, this area has become more and more discrete. This is because finding the right prompt is critical for the model to produce the most accurate answer possible (Brown et al., 2020a; Radford et al., 2025). Another advantage of prompt engineering is that there is no need to retrain the model (Schick & Schütze, 2020).

The foundations of Prompt engineering go back to the early days of natural language processing (NLP). Initially, these methods were simple structures based on keyword matching. The user would search for specific words and get the required output as a result of this search. Over time, as statistical methods and n-gram approaches were replaced by neural network-based methods, the inputs needed to be increasingly structured.

With the introduction of transformer architectures and the attention mechanism, correct prompt design has become increasingly important when establishing long contextual relationships. Finally, with models such as GPT and BERT, prompt engineering has become a technique that needs to be developed in order to get the right answers. Today, instead of manual query generation techniques, reinforcement learning methods can also perform automatic prompt generation (Muktadir, 2023).

### *2.3.1 Prompt Design Rules*

Prompt design has a direct impact on the output of the model. There are two basic elements in a prompt. The goal is set for the output and the context of the prompt. The other prompt elements are related to extending the context of the prompt. In this context, as long as certain rules are followed in prompt design, a correct prompt output can be obtained. These rules are indispensable regardless of the type of prompt. An optimal prompt should have these characteristics (P. Liu et al., 2023).

Here, the prompt should clearly state what the model should do. Vague, ambiguous, or open-ended instructions may lead the model to produce incorrect or irrelevant responses. Clarity enables the model to grasp the task boundaries correctly. For example, instead of saying "Write a story", it would be better to say "Write a story of 300 words, about the Middle Ages, with a happy ending".

Sufficient context must be provided for the model to produce an accurate and meaningful response. Context helps the model to understand the topic, the goal and the relevant background. When context is not provided, the model may act on its own invented data. For example, instead of saying "Write an analysis", it would be more

accurate to say "Analyze table I gave you". Because the model cannot analyze without knowing what to analyze about.

Prompt should be as task specific as possible. General statements can lead the model to be too broad or to cover the wrong areas. Specificity narrows the output of the model to directly serve the purpose. For example, if a table is to be created, it is necessary to know what the table will be about and the format of the output.

The Prompt should contain as much information as is necessary to describe the task, avoiding verbiage. Unnecessary details prevent the model from focusing. A concise prompt produces faster and more accurate results. Phrases such as "Please", "Hello, how are you?" are unnecessary and can reduce the quality of the output.

The prompt should contain elements directly related to the task and should be free of distracting, irrelevant, or unnecessary content. Unnecessary information makes it difficult for the model to produce meaningful output. For example, if it is asked for a recipe and asked to write a code at the same time, the quality of the output will decrease.

Table 2.1 shows the prompt rules that should be created appropriately and a brief description of them. By looking at this table, you can see how a correct prompt should be created.

Table 2.1 Prompt design rules and their descriptions

| Rules | Descriptions |
|---|---|
| Clarity | Instructions should be clear and unambiguous. |
| Context | Prompt should contain meaningful context to guide the model to generate a response. |
| Specificity | Prompt should be task-specific and general or vague statements should be avoided. |
| Conciseness | Prompt should not contain unnecessary information and should be short and concise. |
| Relevance | Prompt should be directly linked to the task and free from distracting content. |

### *2.3.2 Prompting Techniques*

Another requirement for effective and purposeful output in LLMs is the use of correct prompting techniques. There are many prompting techniques in the literature. Prompt techniques should be chosen according to the user's demand. Different prompting techniques can exist and be integrated with each other. Each technique has its own advantages and disadvantages. In this context, factors such as token consumption, acceptable amount of hallucinations, expected text length, and format of the output are important in choosing the right prompt technique. The process of choosing the right prompt technique is intuitive, but knowing the different techniques helps to identify the right method and to fine-tune it appropriately.

#### *2.3.2.1 Standard Prompting*

Standard prompting is to deliver the prompt in the simplest possible way. We can call this approach the "Hello World!" of prompt engineering. It is the request in the simplest possible form. The request is sent through the client, and a response is received. No output format or sample is provided. This technique mostly involves prompts made by the end user. It is the default working style of chatbots. It cannot be directly integrated into any system.

#### *2.3.2.2 Zero-Shot Prompting*

Tasks without any examples are called zero-shot prompting. Although the rate of successful production is low, it can be effective in the production of creative texts as it gives the model flexibility in terms of output. This is because the format of the tasks given with zero examples is similar to the structure in the pre-training process. Effective results can be obtained when used in combination with other methods. This method is effective in processes such as defining tasks and giving commands.

When information is requested on a subject, comprehensive information can be obtained. It is a method that prioritizes human preferences. Appeals to the end user. Instructions are given directly by the user. The gradual definition of instructions is

called instruction tuning. It is very difficult to get the most accurate answer in one shot (Wei, Bosma, et al., 2022).

#### *2.3.2.3 Few-Shot Prompting*

Few-shot prompting aims to exploit the in-context learning capabilities of LLMs by adding a small number of examples to the inputs sent to produce the desired outputs for specific tasks. The aim of this method is to achieve high performance for a small number of examples by using the model's existing general knowledge and context learning ability. In this method, the model is given one or more examples to improve the quality of its outputs (Victoria Lin et al., 2025).

Few-shot prompting is different from zero-shot prompting, where the model runs with zero examples, i.e., no examples are given. In zero-shot prompting, only instructions are given. In few-shot prompting, examples are given, and the goal is for the model to recall the necessary information from similar experiences it has learned in the past (P. Liu et al., 2023). As highlighted in Figure 2.10, few-shot prompting is more effective in high-parameter models.

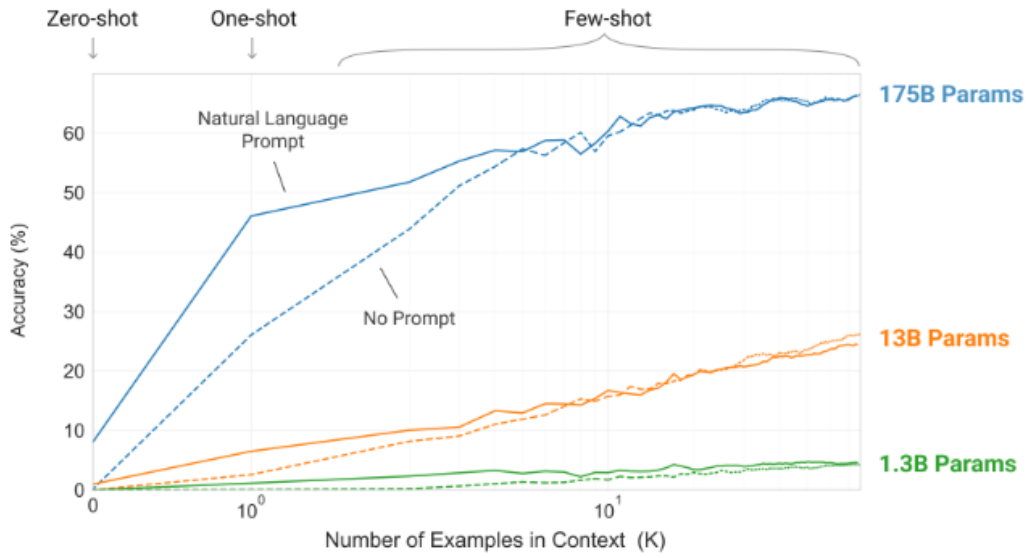



Figure 2.10 Increase in accuracy rate of in the In-Context Learning phase (Brown et al., 2020b)

In this method, it is important that the examples are carefully selected. The examples are given in the form of input-output. This context ensures that the desired type of output is obtained for a given input (Y. Li, 2023). The selection of examples allows the model to generalize in a more specific area by establishing a relationship with examples of similar structure that the model has seen before. As with other methods, this method can be synthesized using different techniques.

The success of Few-shot prompting has a direct correlation with the selection of the right examples. For example, if the model is to perform sentiment analysis, it is important to correctly identify and parse sentiment labels. This method works not only for text generation or classification, but also for text translation, summarization, sentiment analysis, and even for visual output. It also helps to achieve high accuracy in multimodal systems.

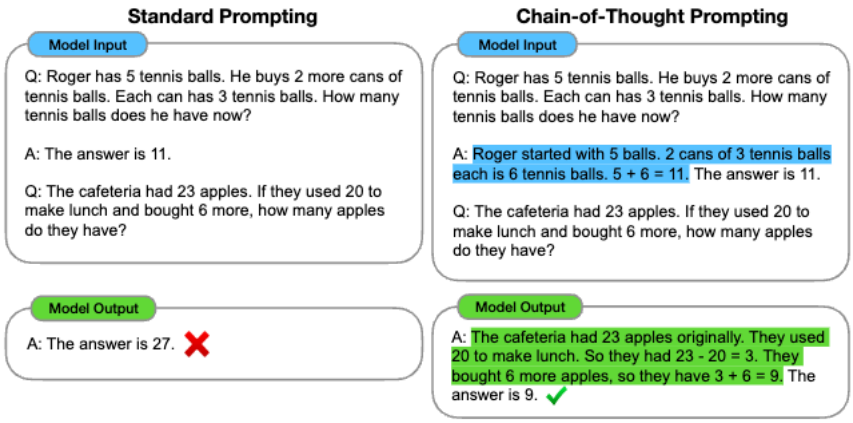



Figure 2.11 CoT prompting enables LLMs to handle complex arithmetic, common sense, and symbolic reasoning tasks (Wei et al., 2022)

The disadvantages of this method are, first and foremost, that the correct sample selection plays a critical role. Wrong or irrelevant examples can significantly reduce the output of the model. If too many examples are given, the model may experience a kind of overloading between examples, and the consistent responses may drop significantly. Also important in this method is the dataset on which the model is trained and the way it is trained (B. Chen et al., 2023). In Figure 2.11, a math question is asked

of the language model. In the example in the figure, the CoT prompting method achieved higher success.

The disadvantages of this method are that if thought processes are not defined correctly in few-shot learning, incorrect thought processes may occur. It is not effective with small models. Computational cost is high because the model thinks longer. If a mistake is made in one step, other steps will follow, and hallucinations will occur. Identification of thought processes in some complex problems may require expertise (Schulhoff et al., 2024).

#### *2.3.2.4 Perspective Prompting*

Perspective prompting is one of the most effective techniques of language models. This technique is of great interest not only in artificial intelligence but also in psychology, sociology, and cognitive sciences in general. In this technique, the model takes the point of view of a particular system or person (Persona Assigned Language Models) and responds accordingly. It is inspired by the theory of mind. Figure 2.12 is an example of how the same situation can be seen from two different perspectives.

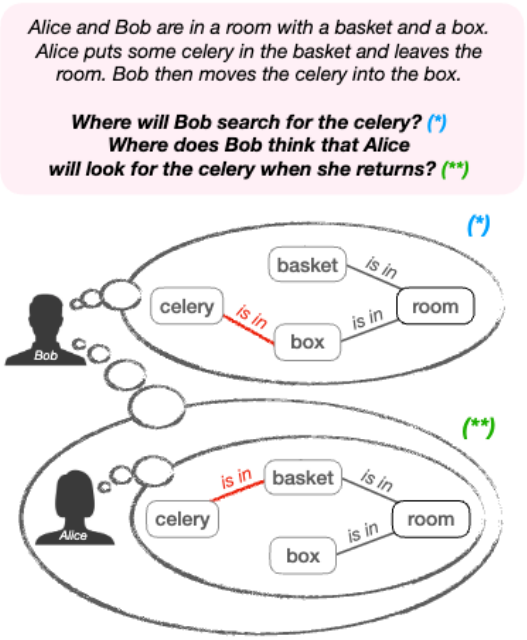



Figure 2.12 Perspective thinking visualization (Sclar et al., 2023)

This technique is the transfer of the human ability to imitate other people's thinking processes to the language model. It is assumed that AI systems can develop different solution strategies by imitating the ability of humans. Likewise, the ability to take perspective can reduce prejudices and enable more empathetic behavior. It can not only imitate the perspective of a person, but also the way that person conveys information and knowledge (Wilf et al., 2023). In this respect, it provides more humane outputs. It reduces toxicity and enables more creative outputs when necessary (Xu et al., 2024). As with other methods, it can be combined with the CoT technique. It can synthesize multiple perspectives and make the creative solution process unique. It can create a common language by taking the role of different professional groups in interdisciplinary studies.

*2.3.2.5 CoT Prompting*

CoT is based on enabling language models to generate the intermediate steps of thinking before reaching a conclusion. While classical methods produce direct answers, CoT makes this process explicit. This both increases the success rate of the model and adds transparency to the output. It allows us to get an idea of how the outcome is produced, rather than seeing the outcome directly.

This is achieved by transferring the human way of thinking to the language model. It is often used with the prompt "think step by step". It breaks down complex problems into steps through reasoning. It breaks down complex problems into steps through reasoning. If supported by few-shot learning in processes such as solving mathematical problems, the probability of getting the right result may increase. Even if the thought process is not directly defined, the model itself can generate an automatic thinking step by step (Raffel et al., 2019; Wiegreffe et al., 2021).

This method allows the behavior of the model to become more explainable through step-by-step analysis. It is effective in mathematical problems, symbolic operations, problems that require common sense. All LLMs can use this method effectively. The image below shows an example of how this process works.

#### *2.3.2.6 Thread-of-Thought Prompting*

Unlike CoT, Thread-of-thought (ThoT) is achieved by creating a pipeline of different types of reasoning models. It can be said to be a method that builds on CoT. Whereas CoT is step-by-step thinking, ThoT creates a logical flow between different chains of thought that think step-by-step. Abstracting different reasoning steps creates a broader perspective on the reasoning process (Zhou et al., 2023).

Theoretically, this approach is based on transforming human reasoning processes into different mental narratives, transforming reasoning into a process of inter-opinion reasoning. In this context, the model is not just a chain of thought. It can also create an ongoing process of thought development (Gou et al., 2024).

As a method, it is created by feeding different reasoning models from each other's outputs. In this way, intermediate processes become more organized. Successive narratives feed on each other, allowing different perspectives and thought processes to develop each other. It significantly increases interpretability. Approaches such as Graph-of-thoughts (GoT), Tree-of-thoughts (ToT) are formed by transferring different chains of thought into a specific graph structure. It creates an infrastructure for Multi-Agent LLM systems.

The biggest risk in these models is overgeneralization. Overgeneralization reduces the clarity of the information, even if it ostensibly reduces the amount of hallucination. As the thread of thought lengthens, abstraction increases and starts to turn into a hypnotic narrative. The specific information requested starts to be presented in a coherent but generalized way, rather than in detail. However, it is a promising method in the long run (Turpin et al., 2023).

### ***2.3.3 Solo Performance Prompting (SPP)***

SPP is a synthesis of perspective prompting and CoT techniques. As with other techniques, it can be developed by supporting it with few-shot prompting. It is based on the idea that personas from different areas of expertise meet for a common solution. They discuss the problem in a certain number of rounds. They can improve each other's

ideas or correct each other's mistakes. The starting point is based on the idea of brainstorming.

People can brainstorm on a single model or on the interaction of different models. With self-evaluation, the model corrects its own mistakes in a loop. It shows similar weaknesses to ThoT in terms of vulnerability. With the right number of iterations and the right characters, it can be used as an interesting but effective method (Z. Wang et al., 2023). Figure 2.13 is a diagram of the SPP method used for story writing in this study.

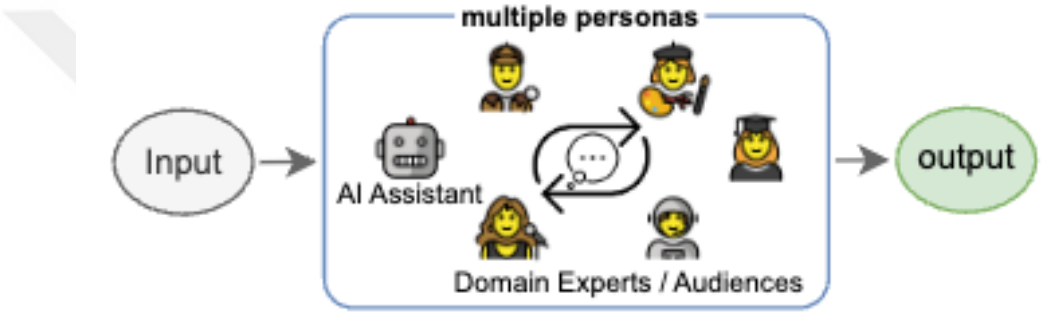



Figure 2.13 Schematic illustration of SPP (Z. Wang et al., 2023)

## 2.4 Multi-Agent LLM Systems

Single-Agent Systems are single-objective LLM systems that independently perceive inputs and display results. In simple automation tasks, such systems make decisions based on their own characteristics (L. Wang et al., 2024). Individual agents are equipped with attributes and capabilities to fulfill specific task particles. The perception dimension is shaped according to the inputs that the agent receives from outside. These inputs can be sensor data, user input, or camera images. The advantage of single agents is the dedication to a specific task. Since they perform only one task, they can provide fast and accurate responses to a specific task.

Furthermore, when the task is multifaceted and complex, these agents may be insufficient. This is where MAS comes into play. MAS are complex systems consisting of multiple single agents that interact with each other (Hu et al., 2021). The advantage

of MAS is that they can have a distributed decision-making structure and can be adapted to larger systems by sharing information with each other.

With the rapid development of LLMs in recent years, LLM-based MAS have been developed. These systems can communicate more complexly and flexibly by utilizing their superior ability to understand and generate natural language. They can facilitate production processes by making the interaction between non-LLM agents closer to human interaction. In this way, they can handle more complex and dynamic tasks (He et al., 2024).

LLM based MAS are usually integrated into subsystems that can be expressed in human language. With the created workflows, agents have a life cycle within the complex task. This cycle can include agent creation, perception, reasoning, action execution and self-learning (X. Li et al., 2024).

### *2.4.1 Collaboration Types*

In LLM based MAS, the types of collaboration affect the overall functioning of the system in terms of task accomplishments and outputs. Different types of collaboration determine how agents communicate with each other, how they share information and how they contribute to decision-making. The types of cooperation used in these systems can be categorized into three types: cooperation, competition, coopetition (Lu et al., 2024). Figure 2.14 visualizes three different types of collaboration.

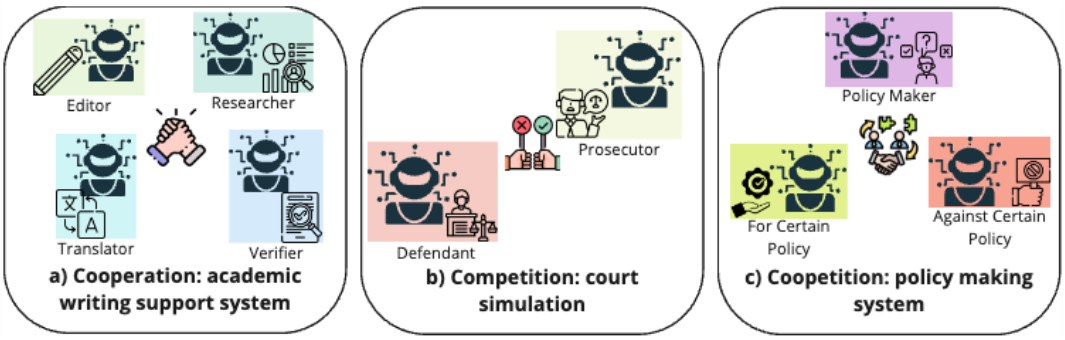



Figure 2.14 Illustration examples of collaboration types (Tran et al., 2025)

#### *2.4.1.1 Cooperation*

It is the most widely used and most researched type of collaboration in LLM-based MAS. In this method, agents share information among themselves towards a common goal. They divide tasks and learn from each other to achieve better results. In LLM-based MAS, collaboration is realized through mechanisms such as combining and sharing information and making decisions.

For example, different agents can perform inter- or intra-level task planning by distributing processes such as memory, information retrieval, and planning to different specialized agents. This method has the advantages of high capacity to achieve a common goal, agents can complement each other's shortcomings, and individual mistakes can be corrected with a collective mind (Han et al., 2024).

#### *2.4.1.2 Competition*

Competition is not as widely used as cooperation. However, it is still important. Because when agents compete with each other, not only will they develop better solutions, but they will also be more likely to correct each other's mistakes. Competition is often used directly in debate-based protocols. In a decision-making process, agents try to refute each other's decisions instead of supporting them. In the end, the most convincing joint solution is reached by an impartial arbiter who monitors the debate. By mimicking different human perspectives, more ethical decisions can be made and biases can be reduced, allowing for faster discovery of missing or incorrect information. The results are likely to be more validated. Results are obtained through a more critical process (Zhao et al., 2024).

#### *2.4.1.3 Coopetition*

It is a combination of cooperation and competition. In complex structures, agents cooperate on some tasks and compete on others. This structure has not yet been extensively studied. Furthermore, in large systems, this structure can be used in some

established subsystems. It can increase the variety of solutions or contribute to the development of creative alternatives (Abdelnabi et al., 2024).

### *2.4.2 Communication Structures*

Communication structures are the structures that are determined according to the requirements of the system and determine how the actors of the system interact with each other. How cooperation is carried out is determined through these structures. Different communication structures can be effective for different applications. In this section, communication structures are analyzed under four main headings. Figure 2.15 shows a simple illustration of the four methods.

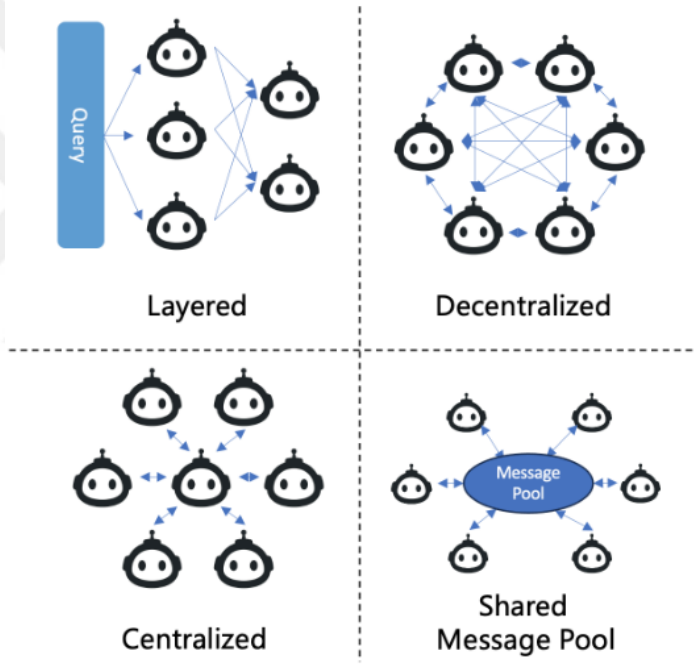



Figure 2.15 Agent communication structures (Guo et al., 2024)

#### *2.4.2.1 Layered Structure*

Layered structure is a structure where there is a hierarchy between agents. Different layers have different roles, functions and authorizations. Usually the agents in the upper layers are decision makers. Those in the lower layers are used for information gathering and execution. Layered communication structure is widely used in LLM

based MAS. It is especially preferred for progressive information processing and scheduling of complex tasks (Z. Liu et al., 2024).

#### *2.4.2.2 Centralized Structure*

In a centralized structure, the system is managed by a single central agent. The central agent is at the center of the information flow. The central agent directs communication, assigns tasks, and provides information integration. It has advantages such as centralized structure, fast decision making, low communication cost. It is an effective method and widely used in small-scale MAS environments (Y. Chen et al., 2024).

#### *2.4.2.3 Distributed Structure*

These are structures in which agents communicate peer-to-peer and without a central point of control. In this structure, each agent can communicate directly and independently with the others. These structures provide flexibility and fault tolerance. Effective in large-scale MAS applications. It is costly and requires complex communication protocols and well-defined coordination mechanisms. The system has high vulnerabilities (Aher et al., 2023).

#### *2.4.2.4 Shared Message Pools*

This method, introduced by MetaGPT, is a model that uses a common messaging space or a common shared data store. Agents can leave messages in the common repository, and other agents can read them. This approach is easy to implement and allows synchronous and asynchronous communications. Agents do not need to know each other's internal state, but it can pose challenges in terms of routing data traffic correctly, scheduling messaging, and scalability (Hong et al., 2023).

# CHAPTER THREE
# METHODOLOGY

In the first section of this chapter, a flowchart of the general operation of the system is presented. The second section details the operations on the dataset that is used as the basis for synthetic complaint generation. Then, the generation of synthetic complaints and the examples of prompts used at this stage are presented.

The third section explains how the GPT, Claude, and Gemini models are implemented in the Python programming language.

The next section describes how prompt techniques are implemented, respectively, Standard Prompting, CoT, SPP, and the stablishment of the SPP + ThoT pipeline, which is a MAS based structure, is explained.

Subsequently, evaluation process of these complaints is detailed and on what criteria the tests are based. Last section of this chapter, provides a simple user interface to present the techniques developed and the models used to the end user.

## 3.1 System Overview

Figure 3.1 shows the general flow of the system. To summarize the system flow, the Federal Communications Commission (FCC) data (FCC, 2024) was used for data preparation. Afterwards, a preprocessing was performed on this dataset. Afterwards, two columns were created, and consciousness and toxicity values were given. A synthetic complaint dataset was created based on these values. During the preparation of the response, four different techniques were used: Standard Prompt, CoT, SPP, and Hybrid approach. These stages were tested separately in all models. The outputs generated in the evaluation phase were evaluated using six different evaluation criteria of the evaluation model, and the numerical results obtained were compared with each other.

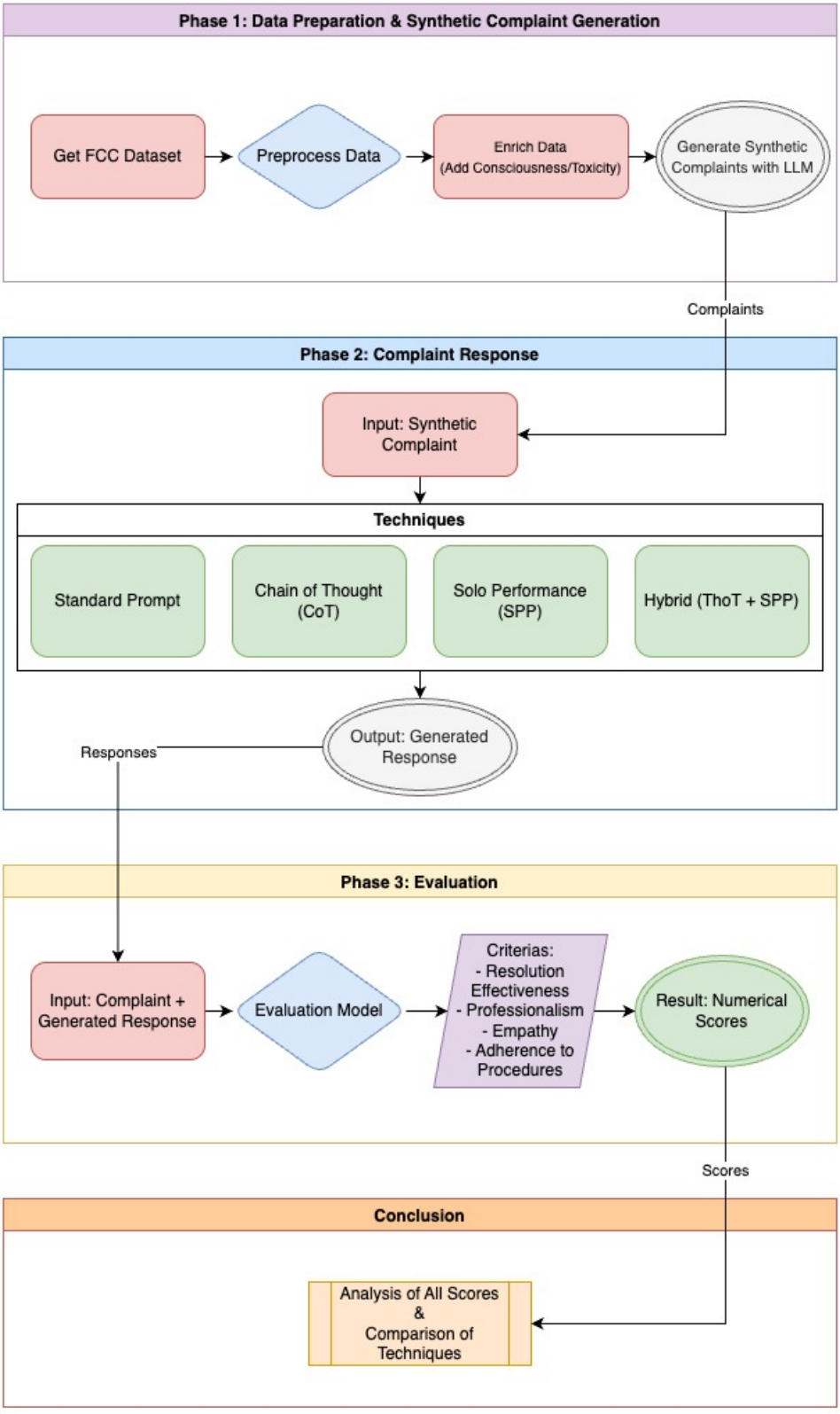



Figure 3.1 System flowchart

## 3.2 Complaint Generation

This stage is the creation of the test data to be used in the study. In order to determine how the models and techniques perform in the relevant scenarios, a real data pool was used. This dataset is based on the FCC's Customer Complaint Dataset, which contains more than 3 million complaints. This dataset was chosen because it is accessible, continuously updated, and the types of complaints and industry types are regularly updated.

### *3.2.1 Data Preparation*

In the initialization phase, 'Not a Number (NaN)' values were removed from the data. Since our aim is to deduplicate the service types here, they were deleted from the cleaned data. Columns containing unknown services were deleted. As a result, we were left with 26 different service types. The identified service types were manually assigned an industry type. The purpose of assigning an industry type is that the same service types in a certain sector will follow different solutions. A separate table was created from these service types. The table can be accessed below.

After that, a customer database of 100 people was created. These are our synthetic customers, and the table where we will create synthetic complaints. The people in this table are assigned toxicity and consciousness levels, respectively.

Customers were given a random toxicity level between 1 and 5 to represent the mood of their complaints. This is because not all complaints are intended to have a neutral tone. The aim of this study is to generate a complaint taking into account the customer's emotional state. In other words, it is important to give the appropriate response according to the customer's emotional state.

In the last stage of the data preparation process, a random level of consciousness was assigned, similar to the toxicity level. The aim here is that each customer type has a different level of consciousness, and the automatic responses to be given will be based on this level of consciousness. This is because a customer who has detailed

knowledge about the service or product he/she is buying and an unconscious customer should not have the same answer. Table 3.1 includes the created industries and their related service types.

Table 3.1 Industry classification by service types

| Industry | Service Type |
|---|---|
| Mortgage & Lending | Mortgage Service |
| Card Services | Credit card or prepaid card |
| Retail Banking | Checking or savings account |
| Student Lending | Student loan |
| Auto Finance | Vehicle loan or lease |
| Personal & Short-Term Lending | Payday loan, title loan, or personal loan |
| Payment & Digital Currency | Money transfer, virtual currency, or money service |
| Banking | Bank account or service |
| Costumer Lending | Costumer Loan |
| Credit Reporting | Credit reporting |
| Prepaid Card Services | Prepaid card |
| Other Financial Services | Other financial service |
| Payday Lending | Payday loan |
| Credit Card Issuance | Credit card |
| Money Transfer | Money transfers |
| Virtual Currency | Virtual currency |
| Internet Services | Broadband Internet Service |
| Mobile Communications | Wireless Telephone Service |
| Fixed-Line Communications | Wired Telephone Service |
| Cable TV | Cable Television Service |
| Satellite TV | Satellite Television Service |
| VoIP Services | Voice over IP (VoIP) Service |
| Hardware | Computer Hardware |
| Software & Online Services | Software and Online Services |
| E-commerce & Retail | E-commerce/Retail |
| Health Insurance | Health Insurance |

### *3.2.2 Synthetic Complaint Generation*

Based on the data in the table, various complaint texts were generated using GPT-4o. At this stage, the role of the customer was assigned to the language model. At this stage, a complaint database was created using large language models, taking into

account the relevant service type, industry, toxicity level, and awareness level of the customer. In order for each complaint to be unique, the temperature value was set to 0.7 on the API and creative text generation was brought to the global optimum value.

The relevant code can be seen as follows. Service type, industry, consciousness and toxicity information are taken as constructor. generateAsisstantPrompt() method shows the system prompt that the language model will use. Here, an example problem is defined with few-shot prompting. Then, with the getUserPrompt () function, the data from the relevant sale is added to the last prompt. The corresponding class structure can be found in Figure 3.2.

```
class complaint_generator:
    def __init__(self, type='', industry='', consciousness=5, toxicity=1):
        self.type_of_service = type
        self.industry = industry
        self.customer_consciousness = consciousness
        self.customer_toxicity = toxicity
        self.assistant_prompt = '' # assistant prompt
        self.generateAsisstantPrompt()
        self.user_prompt = ''

    def generateAsisstantPrompt(self):
        self.assistant_prompt = 'You are a customer of a brand. Your task is to make a complaint about your assigned a
        self.assistant_prompt += 'If its a cell phone, you should provide brief information about the problem you are
        self.assistant_prompt += 'The answer you produce should be related to the customers level of awareness, toxici
        self.assistant_prompt += 'For example, a 1/5 conscious customer will give you a little detailed information. \
        self.assistant_prompt += '5/5 conscious customers will give you detailed information about the problem they ar
        self.assistant_prompt += '\n'
        self.assistant_prompt += 'Here is example: \n'
        self.assistant_prompt += 'Question: Industry: Social Security \n'
        self.assistant_prompt += 'Problem: Delayed Benefit Payments Customer \n'
        self.assistant_prompt += 'Consciousness: 3/5 \n'
        self.assistant_prompt += 'Toxicity: 4/5 \n'
        self.assistant_prompt += 'Answer: \n'
        self.assistant_prompt += 'I was told that I would receive 2 thousand one-time coupons from the Istanbul Metrop
        self.assistant_prompt += 'It was stated that there would be a message during the day, but there was no message
        self.assistant_prompt += 'I need it very much, it would be good if you could help me. I called 153, they said,
        self.assistant_prompt += 'They said there was a waiting period. \n'
        self.assistant_prompt += 'I need it very much. \n'
        self.assistant_prompt += f'.\n'

    def getUserPrompt(self):
        user_prompt = f' Type of service: {self.type_of_service} \n '
        user_prompt += f'Industry: {self.industry} \n '
        user_prompt += f'Consciousness: {self.customer_consciousness} \n '
        user_prompt += f'Toxicity: {self.customer_toxicity} \n '
        return user_prompt
```

Figure 3.2 Complaint generator code

The Python function that interacts with the OpenAI API is shown below. When the required keys are entered, this function directly returns a response based on the assistant prompt and the user prompt. The text is then retrieved from the required

message. The retrieved text is added to the relevant customer's complaint row in the table. The responder function for ChatGPT can be seen in Figure 3.3.

```
def chatGPT_response(
        gpt_user_prompt,
        gpt_assistant_prompt='',
        temperature=0.7,
        max_tokens=4096,
        frequency_penalty=0.0
    ):
    message=[{
            "role": "assistant",
            "content": gpt_assistant_prompt
        },
        {
            "role": "user",
            "content": gpt_user_prompt
        }]
    response = client.chat.completions.create(model="gpt-4o-2024-08-06", # "gpt-3.5-turbo" "gpt-4"
    messages = message,
    temperature=temperature,
    max_tokens=max_tokens,
    frequency_penalty=frequency_penalty)
    return response.choices[0].message.content
```

Figure 3.3 Python function for interacting with the OpenAI API

The complaints generated for an example hardware failure according to different toxicity and consciousness levels are shown in Table 3.2.

Table 3.2 Synthetic customer complaint matrix by consciousness and toxicity

| | High Toxicity | Low Toxicity |
|---|---|---|
| High Consciousness | "I purchased an RTX 3080 from you last month after extensive research. Despite following all recommended installation steps and updating drivers, the card consistently crashes under load. I've monitored temps, swapped out PSU, and even reinstalled Windows — yet the issue persists. I didn't pay a premium for this level of incompetence. If this isn't resolved immediately, I will escalate the issue legally and publicly. Enough is enough." | "I purchased an NVIDIA RTX 3080 graphics card from your store a month ago. It performed well initially, but recently I've encountered frequent crashes during gaming and video editing. I've attempted multiple solutions: driver updates, clean reinstall, BIOS update, OS reinstall, and ensured temperatures stay within 70-75°C. Despite this, the issue persists. I'd appreciate your guidance on next steps or any warranty options available. Thank you for your support." |
| Low Consciousness | "Your graphics card is trash. I bought some card from you — not sure what model — and now my whole computer keeps crashing in games. It worked for a bit and now it's garbage. This is your fault and I want this fixed NOW. If I don't hear back soon, you'll be sorry!" | "Hi, I bought a graphics card from your store last month, but I'm not sure which one it is. Lately, my computer keeps shutting down when I play games. The screen flickers and sometimes it just turns off. I don't know if this is the card's fault or something else. Could you please help me figure out what's going on? I'd really appreciate it." |

### 3.3 Prompting Framework Architecture

In this thesis, a multi-purpose framework architecture has been created for developing web applications and testing prompts using LLMs. This framework is designed to test prompts systematically, analyze the results, and repeat the process. The architecture is modular, layered, and scalable for testing.

The file structure consists of five main parts: application layer, data layer, output layer, analysis layer, and helper layers. Figure 3.4 below shows an outline of the file structure.

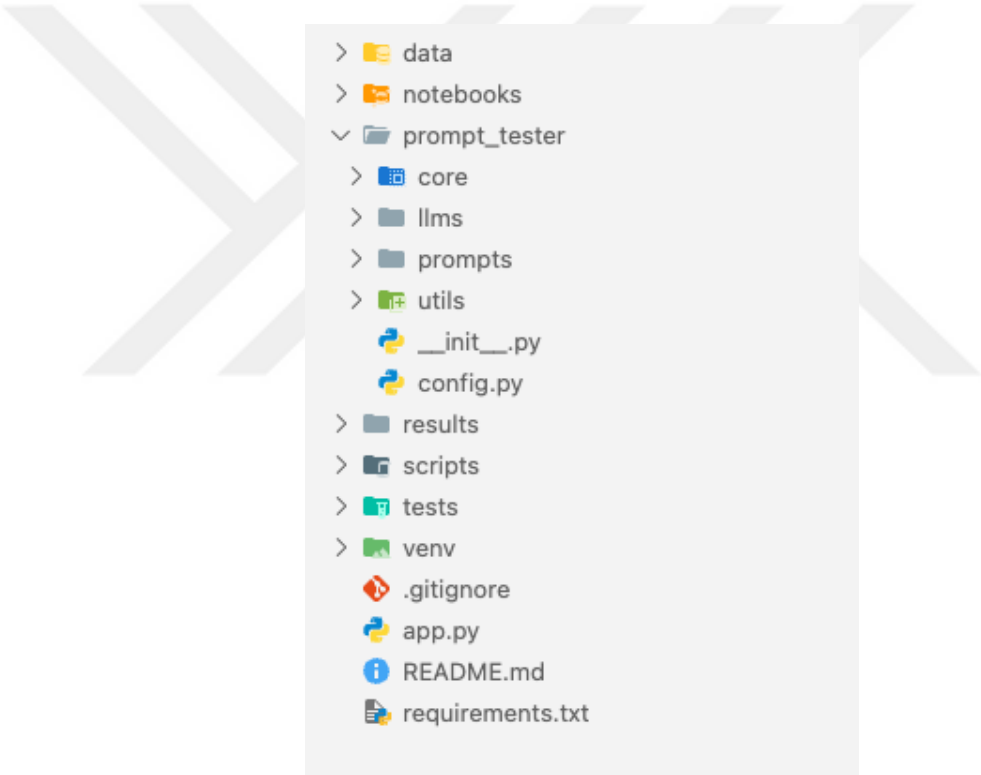



Figure 3.4 File structure of our prompting framework

The '/prompt_tester' clause is the main application layer of the framework. Here, the core section is used to create pipelines for processes where successive tasks are defined, such as thread-of-thought. The '/llms' folder, where the API connection of LLMs is established, abstracts the API connections and makes them manageable from a single center. In the '/prompts' folder, task-oriented prompt definitions are made. Prompt management can be done independently of the code. The '/utils' folder contains

general-purpose utilities that are not directly included in the application. The '/config.py' file contains switches and parameters for related APIs.

The parameters of the tests are set in this file. The '/data' folder contains the raw data. The '/requirements.txt' file contains the versions of the Python libraries in the '/venv' folder. This makes package updates manageable. The '/results' folder contains CSV files in which test results are saved. The CSV files in this section are created automatically using the code in the 'test_config.py' file.

## 3.4 Models Implementations

Three different LLMs were used in this study. These are GPT-4o, Claude-3.5-Haiku, Gemini-1.5-Flash. The implementation of the models used is not exactly the same, but there are minor differences. The implementation of the models is explained below.

### *3.4.1 GPT*

As shown in Figure 3.5, to run GPT-4o, an API key is defined at the beginning. This can be obtained by subscribing to the OpenAI site. Enter the model name in the Completion function. In the System prompt, the prompt describing the technique to be applied in detail is entered. In the User role, the customer's complaint is entered. Output parameters are the maximum number of tokens to be used and the frequency penalty. These values do not have to be entered. However, if they are entered, the parameter can be adjusted.

```
# API key
client = OpenAI(api_key=config.GPT_API_KEY)

messages=[
{"role": "system", "content": system_prompt"},
{"role": "user", "content": customer_complaint}
]

output_temperature = 0.7
max_output_tokens = 450
word_frequency_penalty = 0.5

response = client.chat.completions.create(
  model="gpt-4o",
  messages=message_list,
  temperature=output_temperature,
  max_tokens=max_output_tokens,
  frequency_penalty=word_frequency_penalty
)
```

Figure 3.5 GPT implementation

### *3.4.2 Claude*

Claude integration is very similar to GPT. Unlike GPT, the 'anthropic' library is added. In Claude, the system prompt is added directly to the system parameter, not to the message content. The function 'messages.create' is used in Claude instead of 'chat.completions' used in GPT. Model names and other parameters are the same as in GPT. Figure 3.6 shows the Claude implementation code.

```
import anthropic

client = anthropic.Anthropic(
    api_key=config.ANTHROPIC_API_KEY
)

message_list = [
    {"role": "user", "content": customer_complaint}
]

output_temperature = 0.7
max_output_tokens = 450

response = client.messages.create(
    model="claude-3-sonnet-20240229",
    system=system_prompt,
    messages=message_list,
    temperature=output_temperature,
    max_tokens=max_output_tokens,
)
```

Figure 3.6 Claude implementation

### *3.4.3 Gemini*

The 'genai' library is used in the Gemini implementation. Again, the received API key is entered. In Gemini, the parameters are sent as JSON to the 'generation_config' parameter. To use the last snapshot of the model in Gemini, enter the model name as 'gemini-1.5-pro-latest'. The relevant code can be found in Figure 3.7.

```
import google.generativeai as genai

genai.configure(api_key=config.GCP_API_KEY)

output_temperature = 0.7
max_output_tokens = 450

generation_config = {
  "temperature": output_temperature,
  "max_output_tokens": max_output_tokens,
}

model = genai.GenerativeModel(
    model_name="gemini-1.5-pro-latest",
    system_instruction=system_prompt,
    generation_config=generation_config
)

response = model.generate_content(customer_complaint)
```

Figure 3.7 Gemini implementation

## 3.5 Prompting Techniques

Four different techniques were used to respond to customer complaints. These are standard prompting, CoT, SPP and a combination of ThoT and SPP. These methods build on each other in stages. Methods such as few-shot prompting, zero-shot prompting, and perspective prompting, which are not described in this chapter but used in prompting, are mentioned in the related chapters.

### *3.5.1 Standard Prompting*

This is the most basic prompt example. No methodology has been applied. The reason for including this approach is to measure how the language model responds in the absence of any intervention. The expected response is to suggest a direct solution

or to take on the role of a customer relations person. This simple mechanism is drawn in its simplest form in Figure 3.8.

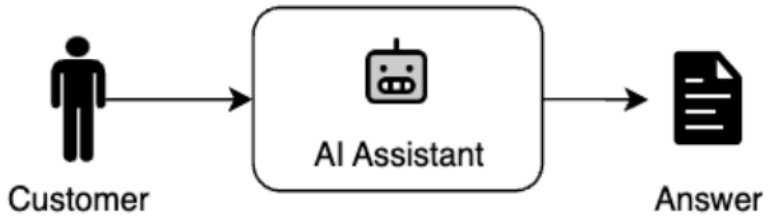



Figure 3.8 Standard prompting diagram

In the vast majority of the experiments she automatically entered the role of web-care staff. In some it said that it could not help. In others, it generated an automated response directly addressing the customer's problem. An example output can be seen in the picture below.

### *3.5.2 CoT Prompting*

Unlike the standard input-output relationship, this method, which allows the language model to respond after going through a thinking process, does not directly respond to the customer complaint, but is asked to first understand the complaint, then develop a solution, and communicate the response. At this stage, a direct customer relations staff role was assigned. In the context of this role, he was asked to approach from the perspective of web-care personnel. She was asked to develop a specific reasoning process using the 'Think step by step' prompt. In this method, few-shot prompting was used.

The process of understanding the complaint is to understand the problem. The solution development process enables the company to create a logical and feasible solution plan for the understood complaint. In the response stage, it is expected to produce a response in the appropriate format according to the customer's emotional state and level of knowledge, and to do so professionally. Figure 3.9 visualizes CoT prompting.

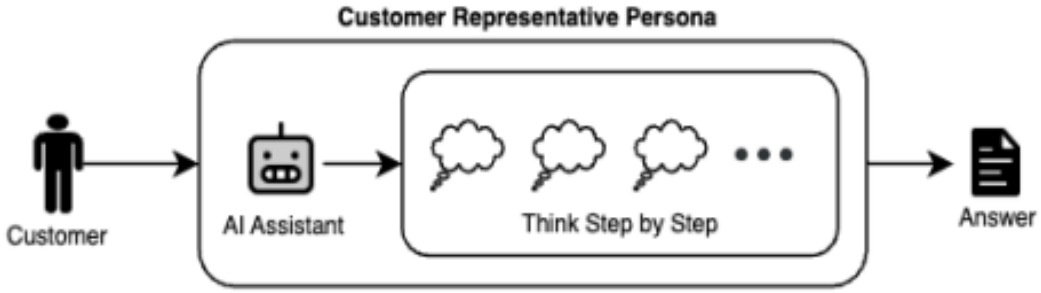



Figure 3.9 CoT prompting diagram

### *3.5.3 SPP*

SPP simulates a meeting environment. It uses a decentralized architecture and cooperation as the collaboration type. In this method, instead of assigning a single persona to the language model, multiple personas are assigned to different areas of expertise. Each area of expertise approaches the problem from its own perspective. They brainstorm among themselves and develop a solution to the problem.

In the first stage, the meeting starts with the customer representative understanding the problem and communicating the problem to the relevant departments. At this stage, a complaint or comment received over the internet is analyzed and transferred to the relevant department in the relevant company. A team list is created according to the type of problem and the list is transferred to the relevant department for a meeting. The diagram of the structure described here can be followed in Figure 3.10.

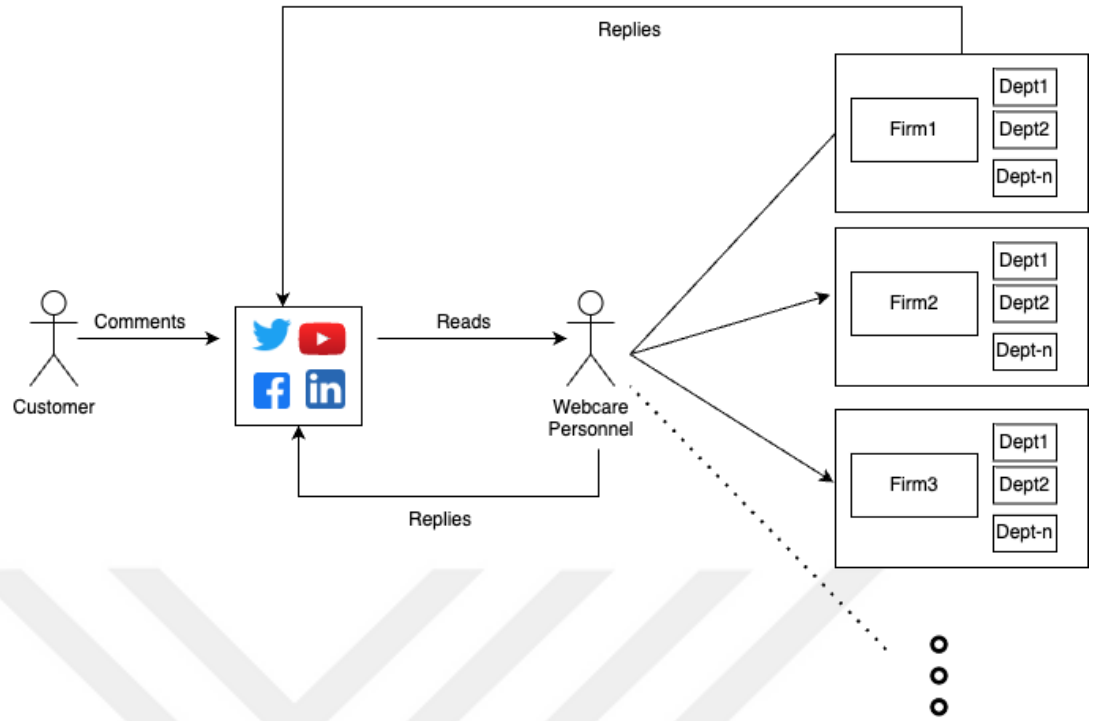



Figure 3.10 The stage at which the customer representative forwards the problem to the relevant department of the company

The team to whom the problem is communicated then initiates a multi-round meeting process between them. The meeting process continues for a certain number of times. When it is the AI assistant's turn, he/she finishes the meeting with the information gathered in the meeting or continues the meeting with a question or comment. As a result, an answer is produced. An example of an SPP made in response to a hardware failure can be understood by looking at Figure 3.11.

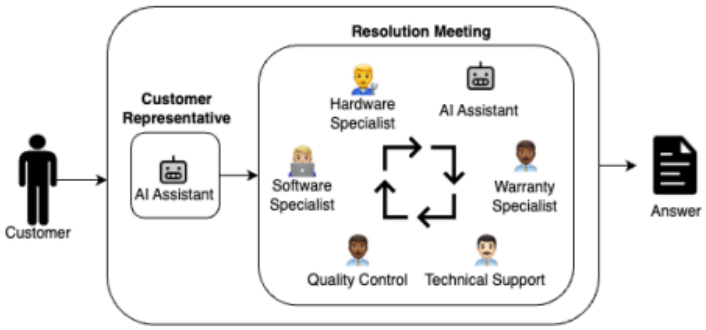



Figure 3.11 SPP visualization

The prompt structure with a sample meeting process created by customer relations staff is summarized in the table below. The steps implemented in the SPP are described step by step in Table 3.3.

Table 3.3 SPP structure

| Stage | Description |
|---|---|
| Role Definition | Assignment of meeting roles by AI assistant |
| Step-by-Step Thinking | The problem is analyzed and understood by the meeting team |
| Team Formation | Defining perspectives of approach to the problem within the team |
| Collaboration (n-step) | Conducting a multi-round discussion and solving the problem in a cyclical manner |
| Final Answer Format | Creating an appropriate response to the customer based on the outcome of the meeting |

### *3.5.4 Multi-Agent Collaboration*

In this method, which is more complex and advanced than the other methods, ThoT, a method in which the multiple personality structure of SPP is used to distribute tasks to different agents, is used. The ThoT method is a method in which agents share successive tasks. The aim of this method is to partition different reasoning processes and thus put the agents into a task-oriented optimization mode. Thus, the processes become more traceable, and logical integrity increases. This process takes place in three main stages.

#### *3.5.4.1 Complaint Analysis Process*

At this stage, the customer representative persona takes charge. It reads and analyzes incoming customer complaints and understands their technical complexity. It then creates a professional-language report to prevent situations such as toxicity in the customer complaint from being directed to the resolution team. In this way, the agent acts as a buffer against toxic complaints and allows the resolution team to enter a more stable resolution process. Figure 3.12 contains an example output diagram.




Figure 3.12 Complaint analysis process

The prompt structure of the related structure is shown in Table 3.4.

Table 3.4 Complaint analysis prompt structure

| Stage | Prompt Template |
|---|---|
| Role Definition | You are an experienced Customer Support Manager. |
| Task | Analyze the following customer complaint, identify the root cause, and suggest the appropriate department to handle it. |
| Context | Here is complaint: *<customer_complaint>* |
| Steps | 1. Identify the main issue in the complaint.<br>2. Classify the type of complaint.<br>3. Determine the customer's emotional tone.<br>4. Create resolution team.<br>5. Suggest the next step for resolution. |
| Output Format | Provide the output as a JSON object with the keys: complaint_report and list_of_participants." |

### *3.5.4.2 Brain-Storming Phase*

At this stage, an SPP process is carried out based on the grievance report and participant list created in the previous stage. The designated expert personas come together, accompanied by a meeting manager. After this stage, a report is transferred to the customer representative. Its place in the flow is shown below. Figure 3.14 shows the SPP phase of the method described here.

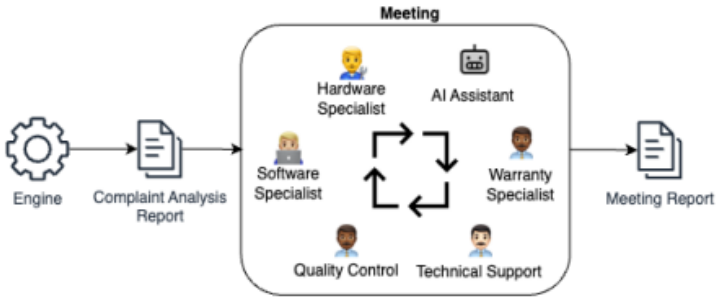



Figure 3.13 Brain-storming step in ThoT

*3.5.4.3 Response Phase*

Stage described in Figure 3.14 where the result is created according to the meeting report. The meeting report is forwarded back to the customer representative persona. The agent responds to the customer based on the customer's level of knowledge, emotional state, company policies and the output of the meeting. In the response phase, which is the last phase of the SPP + ThoT layered structure, the final output is returned to the customer.

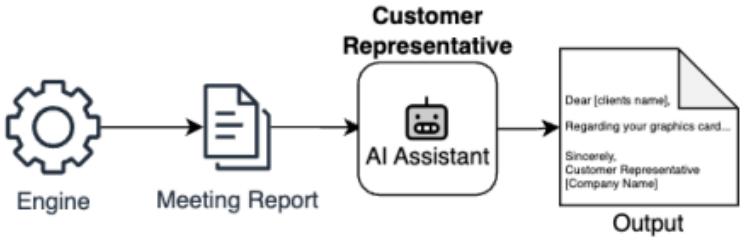



Figure 3.14 Response phase diagram

## 3.6 Evaluation of Answers

In cases such as responding to customer complaints, it is not possible to make a determination of right or wrong directly. Because a complaint management process is not only about solving or not solving the problem. There are multiple factors, such as approach to the customer, management of the solution process, empathy, and reaction time. Therefore, in this study, self-evaluation and cross-evaluation were conducted based on the basic criteria in ISO 9001 Quality Management Standards.

In the prompt design of this phase, a six-stage CoT process was designed. The first stage is to read and understand the response and the customer complaint. The second stage is to score the response according to the criteria: resolution effectiveness, professionalism, empathy, adherence to procedures, and to understand why the response was evaluated in this way. This aims to produce a more explainable answer based on the output. The scores were then summed, and the respondent was asked to give a decimal score between 1 and 0. Table 3.5 details the evaluation phase.

Table 3.5 Evaluation of the answers table

| Step | Prompt Template |
|---|---|
| Complaint Analysis | Read the Customer Complaint and Response |
| Evaluation | Evaluate Against Each Criterion: (Resolution Effectiveness, Professionalism, Empathy, Adherence to Procedures) |
| Explanation | For each score, provide a concise explanation for why that score was given based on observable factors. |
| Scoring | Combine the scores into a total of 100, providing an overall conclusion of the effectiveness of the response. |
| Suggestions | Suggest specific improvements if necessary |
| Output Formating | Write the score in this format: [[0.XX]] |

All these steps were done through Claude, Gemini and GPT. The aim here is to avoid biases in the self-evaluation of the models. Each model was asked to evaluate itself and other models separately. GPT evaluations are available in Appendix 1, Claude evaluations in Appendix 2, Gemini evaluations in Appendix 3.

## 3.7 Application Interface

A simple web interface has been developed to concretize the techniques used here. The application is built to incorporate the methods and language models used in the study. Simply enter the user complaint in the 'Your Complaint' section. Then one of the prompt techniques used in this study is selected. Finally, the user is asked to choose between Gemini, ChatGPT and Claude. After the response is generated, the response to the customer complaint can be seen on the right side. Figure 3.16 shows the user interface created.

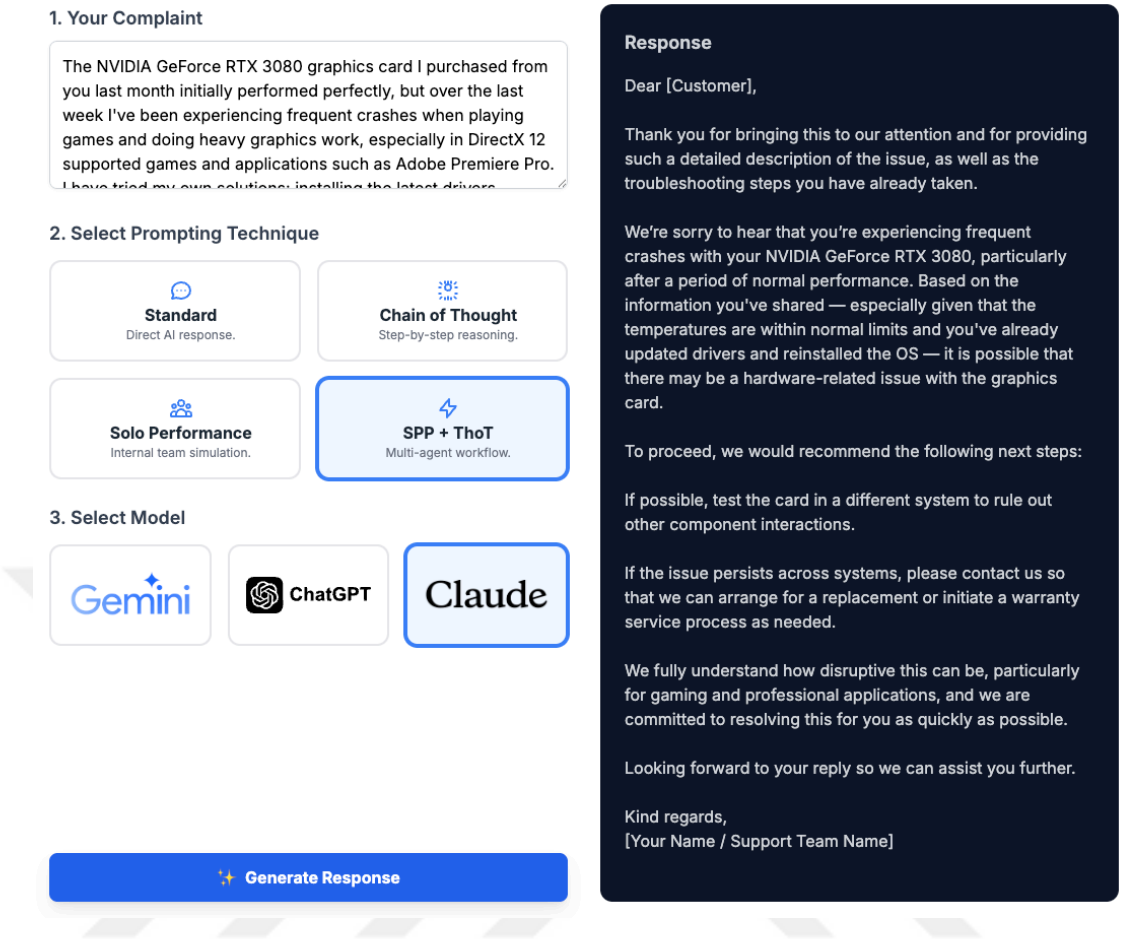



Figure 3.15 Web application user interface

# CHAPTER FOUR
# RESULTS

This chapter presents the results of self-evaluation and cross-evaluation of customer complaint responses based on ISO 9001 standards. The results are evaluated in two aspects: the results of the prompt techniques used and the results of the models used. Below are descriptions of the source and target models.

- **The source model** refers to the scores obtained by its outputs. In other words, it refers to how successful the model's outputs are.
- **The target model** refers to the results of the model when it is evaluated against other models.

Based on the averages of all techniques, Claude-3.5-Haiku has the highest scores in both output generation and output evaluation. It is followed by Gemini-1.5-Flash and GPT-4o. The results can be seen in Table 4.1.

Table 4.1 Overall average scores of models

| Model | Average Source Score | Average Target Score |
|---|---|---|
| Claude-3.5-Haiku | 0.8420 | 0.8340 |
| Gemini-1.5-Flash | 0.8108 | 0.8201 |
| GPT-4o | 0.7809 | 0.7840 |
| **Overall Average** | 0.8113 | 0.8127 |

When we look at the most general results of the prompt techniques used, we see the following results in Table 4.2.

Table 4.2 Overall, the prompt technique results

| Prompt Technique | Average Score |
|---|---|
| Standard Prompt | 0.8042 |
| CoT | 0.7998 |
| SPP | 0.8096 |
| SPP + ThoT | 0.8314 |

In general, the SPP + ThoT prompt technique is seen as the most successful method for responding to customer complaints. It is followed by SPP and standard method. In order to see the performance differences more clearly, we can refer to the graph below in Figure 4.1.

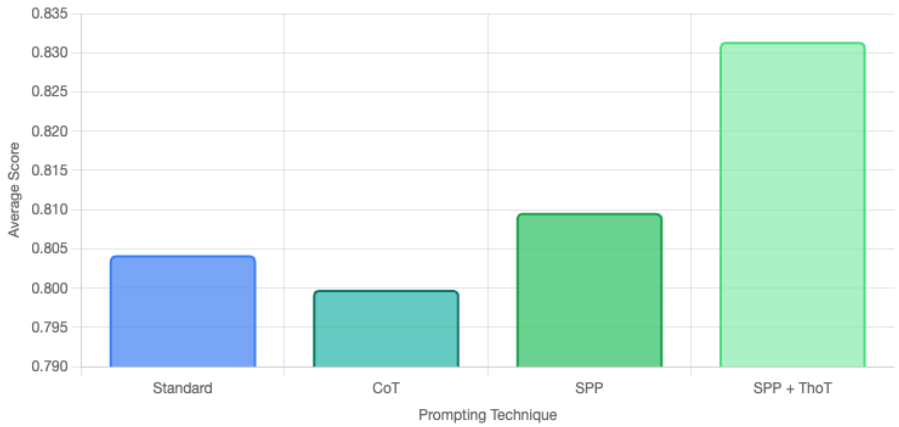



Figure 4.1 Average performance scores of prompting techniques (Higher is better)

The standard deviations of the methods used can be found below. When looking at the standard deviations, lower values are more successful. This graph shows the predictability of the methods and how consistent they are. According to the results, we see the highest consistency in standard prompting, which has the lowest standard deviation. It is followed by the SPP + ThoT method. The related graph can be found in Figure 4.2.

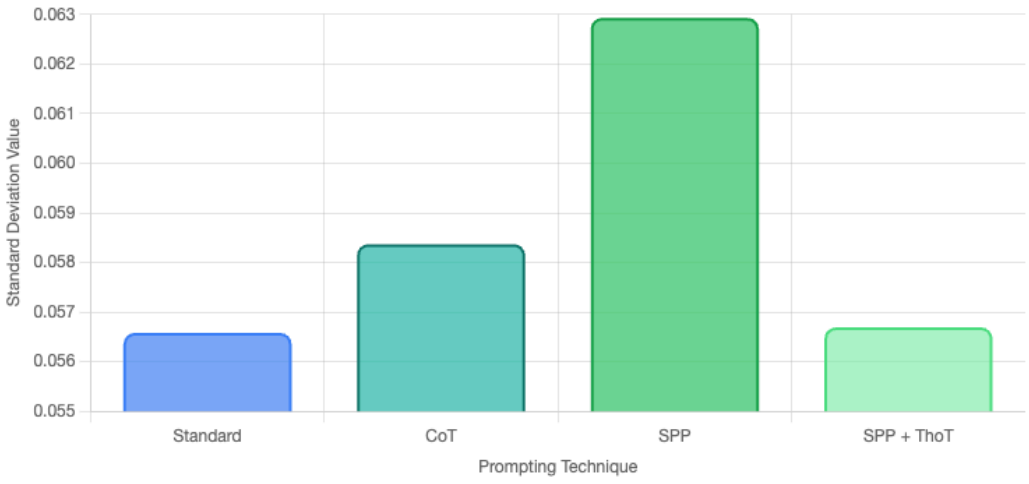



Figure 4.2 Average standard deviation across techniques (Lower is better)

The table below shows the average scores of the models according to the techniques. It can be seen how much success the models achieved in which technique when they were in the source and target roles. In the average scores, the SPP + ThoT method achieved a slightly higher success than the other methods. In the overall average, as seen in Table 4.3, the method using only CoT achieved the lowest success.

Table 4.3 Average scores according to the prompt technique based on the source and target roles of models

| Model | Role | Standard | CoT | SPP | SPP + ThoT |
|---|---|---|---|---|---|
| Claude-3.5-Haiku | Source | 0.84212 | 0.82566 | 0.84355 | **0.85690** |
| | Target | 0.83000 | 0.83083 | 0.82464 | **0.85070** |
| Gemini-1.5-Flash | Source | 0.76864 | 0.78106 | **0.78796** | 0.78584 |
| | Target | 0.77297 | 0.76683 | 0.78949 | **0.80687** |
| GPT-4o | Source | 0.80170 | 0.79264 | 0.79739 | **0.85156** |
| | Target | 0.81483 | 0.80360 | 0.81991 | **0.84203** |

The heatmap according to the source score created with the data from Table 4.3 can be accessed from the Figure 4.3.

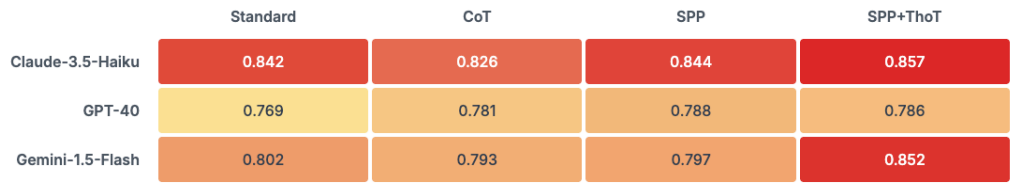



Figure 4.3 Source scores by average heatmap

According to the target score created with the data from Table 4.3, the heatmap can be accessed from Figure 4.4.




Figure 4.4 Target scores by average heatmap

The table below shows the standard deviations of the models in the source and target role separately. GPT-4o has the lowest average standard deviation when in the target role. This shows that GPT-4o produces more consistent responses than the other models. Gemini-1.5-Flash has the highest standard deviation when in the target role. This shows that Gemini-1.5-Flash is more consistent in the target identification process. In fact, it can be concluded that Gemini-1.5-Flash is more rigorous in its evaluation criteria. The performance scores of the models are presented in Table 4.4.

Table 4.4 Standard deviations of models

| Model | Role | Standard | CoT | SPP | SPP + ThoT |
|---|---|---|---|---|---|
| Claude-3.5-Haiku | Source | 0.05735 | 0.06333 | 0.06242 | 0.06403 |
| | Target | 0.06103 | 0.05339 | 0.05641 | 0.06043 |
| Gemini-1.5-Flash | Source | 0.05858 | 0.05907 | 0.07080 | 0.04940 |
| | Target | 0.08799 | 0.76683 | 0.10656 | 0.09805 |
| GPT-4o | Source | 0.05382 | 0.05270 | 0.05557 | 0.05666 |
| | Target | 0.03136 | 0.03092 | 0.04400 | 0.03541 |

# CHAPTER FIVE
# CONCLUSION

In this thesis, an AI-based web-care staff is developed that automatically responds to eWOMs that are important for the reputation of brands using LLMs with different prompt techniques. Four different prompt techniques are compared with each other, and then the performance of these prompt techniques in different language models is analyzed. The SPP + ThoT method produced the highest quality and consistent responses. Among the models, the Claude-3.5-Haiku model achieved the highest scores both in self-evaluation and in the evaluation of other models.

The results obtained in this thesis show the importance of systematic prompting design in LLM-based customer service. In such applications, different prompting techniques should be combined with each other, and different agents should perform different tasks at different stages to increase the success rate. Prompt design is also vital in the transition to multiple agents rather than a single agent doing all the work. Proper prompt design not only increases the success rate in such systems but also increases scalability.

In this context, multi-agent dynamic prompt designs are an important area of study and application for real-time applications and future work. With a properly structured prompting framework, the customer complaint process can generate effective and human-centered responses through LLMs. These developments suggest that in the future, web-care operations can be partially or fully managed by autonomous systems.

## REFERENCES


Abdelnabi, S., Gomaa, A., Sivaprasad, S., Schönherr, L., & Fritz, M. (2024). Cooperation, Competition, and Maliciousness: LLM-Stakeholders Interactive Negotiation. *Advances in Neural Information Processing Systems*, *37*, 83548–83599. https://github.com/S-Abdelnabi/LLM-Deliberation/.

Aher, G. V, Arriaga, R. I., & Kalai, A. T. (2023). *Using Large Language Models to Simulate Multiple Humans and Replicate Human Subject Studies* (pp. 337–371). PMLR. https://proceedings.mlr.press/v202/aher23a.html

Aydın, Ö., & Karaarslan, E. (2022). OpenAI ChatGPT Generated Literature Review: Digital Twin in Healthcare. *SSRN Electronic Journal*. https://doi.org/10.2139/SSRN.4308687

Azaria, A., & Mitchell, T. (2023). *The Internal State of an LLM Knows When It's Lying*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020a). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901. https://commoncrawl.org/the-data/

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020b). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901. https://commoncrawl.org/the-data/

Bruno, A., Mazzeo, P. L., Chetouani, A., Tliba, M., & Kerkouri, M. A. (2022). *Insights into Classifying and Mitigating LLMs' Hallucinations ⋆*. http://ceur-ws.org

*CGB - Consumer Complaints Data | Data | Federal Communications Commission*. (2025). Retrieved June 7, 2025, from https://opendata.fcc.gov/Consumer/CGB-Consumer-Complaints-Data/3xyp-aqkj/about_data

Chen, B., Zhang, Z., Langrené, N., & Zhu, S. (2023). Unleashing the potential of prompt engineering for large language models. *Patterns*, *0*(0), 101260. https://doi.org/10.1016/J.PATTER.2025.101260

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. de O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., … Zaremba, W. (2021). *Evaluating Large Language Models Trained on Code*. https://arxiv.org/pdf/2107.03374

Chen, Y., Arkin, J., Zhang, Y., Roy, N., & Fan, C. (2024). Scalable Multi-Robot Collaboration with Large Language Models: Centralized or Decentralized Systems? *Proceedings - IEEE International Conference on Robotics and Automation*, 4311–4317. https://doi.org/10.1109/ICRA57147.2024.10610676

Chuma, E. L., & Oliveira, G. G. de. (2023). Generative AI for Business Decision-Making: A Case of ChatGPT. *Management Science and Business Decisions*, *3*(1), 5–11. https://doi.org/10.52812/MSBD.63

Cohen, R., Hamri, M., Geva, M., & Globerson, A. (2023). LM vs LM: Detecting Factual Errors via Cross Examination. *EMNLP 2023 - 2023 Conference on Empirical Methods in Natural Language Processing, Proceedings*, 12621–12640. https://doi.org/10.18653/v1/2023.emnlp-main.778

Colliander, J., Dahlén, M., & Modig, E. (2015). Twitter for two: investigating the effects of dialogue with customers in social media. *International Journal of Advertising*, *34*(2), 181–194. https://doi.org/10.1080/02650487.2014.996197

Coombs, W. T. (2002). Assessing online issue threats: issue contagions and their effect on issue prioritisation. *Journal of Public Affairs*, *2*(4), 215–229. https://doi.org/10.1002/PA.115

Curran, S., Lansley, S., & Bethell, O. (2023). *Hallucination is the last thing you need*. https://arxiv.org/pdf/2306.11520

Dash, D., Thapa, R., Banda, J. M., Swaminathan, A., Cheatham, M., Kashyap, M., Kotecha, N., Chen, J. H., Gombar, S., Downing, L., Pedreira, R., Goh, E., Arnaout, A., Morris, G. K., Magon, H., Lungren, M. P., Horvitz, E., & Shah, N. H. (2023). *Evaluation of GPT-3.5 and GPT-4 for supporting real-world information needs in healthcare delivery*. https://arxiv.org/pdf/2304.13714

Ding, Z., Srinivasan, A., MacNeil, S., & Chan, J. (2023). Fluid Transformers and Creative Analogies: Exploring Large Language Models' Capacity for Augmenting Cross-Domain Analogical Creativity. *ACM International Conference Proceeding Series*, 489–505. https://doi.org/10.1145/3591196.3593516/SUPPL_FILE/CC23A-SUB3777-I12.ZIP

Gou, T., Zhang, B., Sun, Z., Wang, J., Liu, F., Wang, Y., & Wang, J. (2024). *Rationality of Thought Improves Reasoning in Large Language Models*. 343–358. https://doi.org/10.1007/978-981-97-5501-1_26

Guerreiro, N. M., Alves, D. M., Waldendorf, J., Haddow, B., Birch, A., Colombo, P., & Martins, A. F. T. (2023). Hallucinations in Large Multilingual Translation Models. *Transactions of the Association for Computational Linguistics*, *11*, 1500–1517. https://doi.org/10.1162/TACL_A_00615/118716/HALLUCINATIONS-IN-LARGE-MULTILINGUAL-TRANSLATION

Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N. V, Wiest, O., & Zhang, X. (2024). *Large Language Model based Multi-Agents: A Survey of Progress and Challenges*. Retrieved June 6, 2025, from https://github.com/taichengguo/LLM

Gupta, P., & Harris, J. (2010). How e-WOM recommendations influence product consideration and quality of choice: A motivation to process information perspective. *Journal of Business Research*, *63*(9–10), 1041–1049. https://doi.org/10.1016/J.JBUSRES.2009.01.015

Han, S., Zhang, Q., Yao, Y., Jin, W., Xu, Z., & He, C. (2024). *LLM Multi-Agent Systems: Challenges and Open Problems*. https://arxiv.org/pdf/2402.03578

He, J., Treude, C., Lo, D., He, J., Treude, C., & Lo, D. (2024). LLM-Based Multi-Agent Systems for Software Engineering: Literature Review, Vision and the Road Ahead. *ArXiv*, arXiv:2404.04834. https://doi.org/10.48550/ARXIV.2404.04834

Hong, S., Zheng, X., Chen, J., Cheng, Y., Zhang, C., Wang, Z., Ka, S., Yau, S., Lin, Z., Zhou, L., Ran, C., Xiao, L., & Wu, C. (2023). *METAGPT: META PROGRAMMING FOR MULTI-AGENT COLLABORATIVE FRAMEWORK*. Retrieved June 6, 2025, from https://github.com/geekan/MetaGPT

Hu, J., Bhowmick, P., Jang, I., Arvin, F., & Lanzon, A. (2021). A Decentralized Cluster Formation Containment Framework for Multirobot Systems. *IEEE Transactions on Robotics*, *37*(6), 1936–1955. https://doi.org/10.1109/TRO.2021.3071615

Jeblick, K., Schachtner, B., Dexl, J., Mittermeier, A., Stüber, A. T., Topalis, J., Weber, T., Wesp, P., Sabel, B. O., Ricke, J., & Ingrisch, M. (2024). ChatGPT makes medicine easy to swallow: an exploratory case study on simplified radiology reports. *European Radiology*, *34*(5), 2817–2825. https://doi.org/10.1007/S00330-023-10213-1/FIGURES/3

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, *55*(12). https://doi.org/10.1145/3571730/ASSET/81F89D0C-B684-4789-A4C7-871B2508418B/ASSETS/GRAPHIC/CSUR-2022-0173-F01.JPG

Jin, Q., Yang, Y., Chen, Q., & Lu, Z. (2024). GeneGPT: augmenting large language models with domain tools for improved access to biomedical information. *Bioinformatics*, *40*(2). https://doi.org/10.1093/BIOINFORMATICS/BTAE075

Karakaya, F., & Barnes, N. G. (2010). Impact of online reviews of customer care experience on brand or company selection. *Journal of Consumer Marketing*, *27*(5), 447–457. https://doi.org/10.1108/07363761011063349/FULL/PDF

Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., Krusche, S., Kutyniok, G., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., … Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large

language models for education. *Learning and Individual Differences*, *103*, 102274. https://doi.org/10.1016/J.LINDIF.2023.102274

Li, J., Chen, J., Ren, R., Cheng, X., Zhao, W. X., Nie, J.-Y., & Wen, J.-R. (2024). *The Dawn After the Dark: An Empirical Study on Factuality Hallucination in Large Language Models*. https://arxiv.org/pdf/2401.03205

Li, X., Wang, S., Zeng, S., Wu, Y., & Yang, Y. (2024). *REVIEW Open Access A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges*. https://doi.org/10.1007/s44336-024-00009-2

Li, Y. (2023). A Practical Survey on Zero-shot Prompt Design for In-context Learning. *International Conference Recent Advances in Natural Language Processing, RANLP*, 641–647. https://doi.org/10.26615/978-954-452-092-2_069

Li, Y., Peng, B., Shen, Y., Mao, Y., Liden, L., Yu, Z., & Gao, J. (2021). Knowledge-Grounded Dialogue Generation with a Unified Knowledge Representation. *NAACL 2022 - 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, 206–218. https://doi.org/10.18653/v1/2022.naacl-main.15

Litvin, S. W., Goldsmith, R. E., & Pan, B. (2008). Electronic word-of-mouth in hospitality and tourism management. *Tourism Management*, *29*(3), 458–468. https://doi.org/10.1016/J.TOURMAN.2007.05.011

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Computing Surveys*, *55*(9). https://doi.org/10.1145/3560815/SUPPL_FILE/3560815-APP.PDF

Liu, Z., Zhang, Y., Li, P., Liu, Y., & Yang, D. (2024). *A Dynamic LLM-Powered Agent Network for Task-Oriented Agent Collaboration*. https://github.com/SALT-NLP/DyLAN.

Lu, J., Pang, Z., Xiao, M., Zhu, Y., Xia, R., & Zhang, J. (2024). *Merge, Ensemble, and Cooperate! A Survey on Collaborative Strategies in the Era of Large Language Models*. https://arxiv.org/pdf/2407.06089

Luo, Z., Xu, C., Zhao, P., Geng, X., Tao, C., Ma, J., Lin, Q., & Jiang, D. (2023). *Augmented Large Language Models with Parametric Knowledge Guiding*. https://arxiv.org/pdf/2305.04757

Menick, J., Trebacz, M., Mikulik, V., Aslanides, J., Song, F., Chadwick, M., Glaese, M., Young, S., Campbell-Gillingam, L., Irving, G., McAleese, N., & contributions, E. (2022). *Teaching language models to support answers with verified quotes*. 2022–2025. https://arxiv.org/pdf/2203.11147

Muktadir, G. M. (2023). *A Brief History of Prompt: Leveraging Language Models. (Through Advanced Prompting)*. https://arxiv.org/pdf/2310.04438

Naveed, H., Ullah Khan, A., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., & Mian, A. (2023). *A Comprehensive Overview of Large Language Models*.

Park, S. M., Georgiev, K., Ilyas, A., Leclerc, G., & Mądry, A. (2023). TRAK: Attributing Model Behavior at Scale. *Proceedings of Machine Learning Research*, *202*, 27074–27113. https://arxiv.org/pdf/2303.14186

Pavlik, J. V. (2023). Collaborating With ChatGPT: Considering the Implications of Generative Artificial Intelligence for Journalism and Media Education. *Journalism & Mass Communication Educator*, *78*(1), 84–93. https://doi.org/10.1177/10776958221149577

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2025). *Language Models are Unsupervised Multitask Learners*. Retrieved June 5, 2025, from https://github.com/codelucas/newspaper

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2019). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, *21*, 1–67. https://arxiv.org/pdf/1910.10683

Roumeliotis, K. I., Tselikas, N. D., & Nasiopoulos, D. K. (2025). Think Before You Classify: The Rise of Reasoning Large Language Models for Consumer Complaint

Detection and Classification. *Electronics 2025, Vol. 14, Page 1070*, *14*(6), 1070. https://doi.org/10.3390/ELECTRONICS14061070

Schamari, J., & Schaefers, T. (2015). Leaving the home turf: How brands can use webcare on consumer-generated platforms to increase positive consumer engagement. *Journal of Interactive Marketing*, *30*, 20–33. https://doi.org/10.1016/J.INTMAR.2014.12.001/ASSET/5A9F101F-F7ED-4497-904E-0237BB71EA1D/ASSETS/IMAGES/LARGE/10.1016_J.INTMAR.2014.12.001-IMG2.JPG

Schick, T., & Schütze, H. (2020). It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners. *NAACL-HLT 2021 - 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, 2339–2352. https://doi.org/10.18653/v1/2021.naacl-main.185

Schulhoff, S., Ilie, M., Balepur, N., Kahadze, K., Liu, A., Si, C., Li, Y., Gupta, A., Han, H., Schulhoff, S., Dulepet, P. S., Vidyadhara, S., Ki, D., Agrawal, S., Pham, C., Kroiz, G., Li, F., Tao, H., Srivastava, A., … Resnik, P. (2024). *The Prompt Report: A Systematic Survey of Prompt Engineering Techniques*. https://arxiv.org/pdf/2406.06608

Sclar, M., Kumar, S., West, P., Suhr, A., Choi, Y., Tsvetkov, Y., & Allen, P. G. (2023). *Minding Language Models' (Lack of) Theory of Mind: A Plug-and-Play Multi-Character Belief Tracker*. *1*, 13960–13980.

Singh Gill, S., Xu, M., Patros, P., Wu, H., Kaur, R., Kaur, K., Fuller, S., Singh, M., Arora, P., Parlikad, A. K., Stankovski, V., Abraham, A., Ghosh, S. K., Lutfiyya, H., Kanhere, S. S., Bahsoon, R., Rana, O., Dustdar, S., Sakellariou, R., … Buyya, R. (2023). Transformative Effects of ChatGPT on Modern Education: Emerging Era of AI Chatbots. *ArXiv*, arXiv:2306.03823. https://doi.org/10.48550/ARXIV.2306.03823

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., & Christiano, P. F. (2020). Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, *33*, 3008–3021.

Thorne, J., Vlachos, A., Cocarascu, O., Christodoulopoulos, C., & Mittal, A. (2018). *The Fact Extraction and VERification (FEVER) Shared Task*. 1–9. https://doi.org/10.18653/v1/w18-5501

Tran, K., Dao, D., Nguyen, H. D., Tran, K.-T., Nguyen, M.-D., & Pham, Q.-V. (2025). *Multi-Agent Collaboration Mechanisms: A Survey of LLMs*. *1*. https://doi.org/XXXXXXX.XXXXXXX

Turpin, M., Michael, J., Perez, E., & Bowman, S. R. (2023). *Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting*.

van Noort, G., & Willemsen, L. M. (2012). Online Damage Control: The Effects of Proactive Versus Reactive Webcare Interventions in Consumer-generated and Brand-generated Platforms. *Journal of Interactive Marketing*, *26*(3), 131–140. https://doi.org/10.1016/J.INTMAR.2011.07.001/ASSET/B7CEFD15-D6A5-4553-8429-5FA1E55EA6A2/ASSETS/IMAGES/LARGE/10.1016_J.INTMAR.2011.07.001-FIG1.JPG

Vermeer, S. A. M., Araujo, T., Bernritter, S. F., & van Noort, G. (2019). Seeing the wood for the trees: How machine learning can help firms in identifying relevant electronic word-of-mouth in social media. *International Journal of Research in Marketing*, *36*(3), 492–508. https://doi.org/10.1016/J.IJRESMAR.2019.01.010

Victoria Lin, X., Mihaylov, T., Artetxe, M., Wang, T., Chen, S., Simig, D., Ott, M., Goyal, N., Bhosale, S., Du, J., Pasunuru, R., Shleifer, S., Singh Koura, P., Chaudhary, V., Wang, J., Zettlemoyer, L., Kozareva, Z., Diab, M., Stoyanov, V., & Li, X. (2025). *Few-shot Learning with Multilingual Generative Language Models*. Retrieved June 6, 2025, from https://github.com/facebookresearch/

Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., Zhao, W. X., Wei, Z., & Wen, J. (2024). A survey on large language

model based autonomous agents. *Frontiers of Computer Science*, *18*(6), 1–26. https://doi.org/10.1007/S11704-024-40231-1/METRICS

Wang, Z., Chu, Z., Doan, T. V., Ni, S., Yang, M., & Zhang, W. (2024). History, development, and principles of large language models: an introductory survey. *AI and Ethics 2024 5:3*, *5*(3), 1955–1971. https://doi.org/10.1007/S43681-024-00583-7

Wang, Z., Mao, S., Wu, W., Ge, T., Wei, F., & Ji, H. (2023). *Unleashing the Emergent Cognitive Synergy in Large Language Models: A Task-Solving Agent through Multi-Persona Self-Collaboration*. Retrieved June 6, 2025, from https://github.com/MikeWangWZHL/

Wang, Z., Mao, S., Wu, W., Ge, T., Wei, F., & Ji, H. (2023). Unleashing the Emergent Cognitive Synergy in Large Language Models: A Task-Solving Agent through Multi-Persona Self-Collaboration. *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2024*, *1*, 257–279. https://doi.org/10.18653/v1/2024.naacl-long.15

Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., & Le Google Research, Q. V. (2022). *FINETUNED LANGUAGE MODELS ARE ZERO-SHOT LEARNERS Closed-Book QA*. https://github.com/google-research/flan.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi Quoc, E. H., Le, V., & Zhou, D. (2022). *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models Chain-of-Thought Prompting*.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi Quoc, E. H., Le, V., & Zhou, D. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems*, *35*, 24824–24837.

Wiegreffe, S., Hessel, J., Swayamdipta, S., Riedl, M., & Choi, Y. (2021). Reframing Human-AI Collaboration for Generating Free-Text Explanations. *NAACL 2022 - 2022 Conference of the North American Chapter of the Association for*

*Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, 632–658. https://doi.org/10.18653/v1/2022.naacl-main.47

Wilf, A., Lee, S. S., Liang, P. P., & Morency, L.-P. (2023). *Think Twice: Perspective-Taking Improves Large Language Models' Theory-of-Mind Capabilities*. https://arxiv.org/pdf/2311.10227

Willemsen, L., Neijens, P. C., & Bronner, F. A. (2013). *Webcare as Customer Relationship and Reputation Management? Motives for Negative Electronic Word of Mouth and Their Effect on Webcare Receptiveness*. 55–69. https://doi.org/10.1007/978-3-658-02365-2_5

Xu, R., Zhou, Z., Zhang, T., Qi, Z., Yao, S., Xu, K., Xu, W., & Qiu, H. (2024). *Walking in Others' Shoes: How Perspective-Taking Guides Large Language Models in Reducing Toxicity and Bias*. https://arxiv.org/pdf/2407.15366

Ye, H., Liu, T., Zhang, A., Hua, W., Jia, W., & Lab, Z. (2023). *Cognitive Mirage: A Review of Hallucinations in Large Language Models*. Retrieved June 4, 2025, from https://github.com/hongbinye/

Yong, L. P. X., Tung, J. Y. M., Lee, Z. Y., Kuan, W. Sen, & Chua, M. T. (2024). Performance of Large Language Models in Patient Complaint Resolution:Web-Based Cross-Sectional Survey. *Journal of Medical Internet Research*, *26*(1), e56413. https://doi.org/10.2196/56413

You, Y., Vadakkepatt, G. G., & Joshi, A. M. (2015). A meta-analysis of electronic word-of-mouth elasticity. *Journal of Marketing*, *79*(2), 19–39. https://doi.org/10.1509/JM.14.0169/SUPPL_FILE/AMETAANALYSISOFELECTRONIC.PDF

Yu, W., Zhang, Z., Liang, Z., Jiang, M., & Sabharwal, A. (2023). *Improving Language Models via Plug-and-Play Retrieval Feedback*. https://arxiv.org/pdf/2305.14002

Yuan, W., Neubig, G., & Liu, P. (2021). BARTScore: Evaluating Generated Text as Text Generation. *Advances in Neural Information Processing Systems*, *34*, 27263–27277. https://github.com/neulab/BARTScore,

Zhao, Q., Wang, J., Zhang, Y., Jin, Y., Zhu, K., Chen, H., & Xie, X. (2024). *CompeteAI: Understanding the Competition Dynamics of Large Language Model-based Agents*.

Zhou, Y., Geng, X., Shen, T., Tao, C., Long, G., Lou, J.-G., & Shen, J. (2023). *Thread of Thought Unraveling Chaotic Contexts*.

# APPENDICES

## Appendix 1: Evaluation results of GPT-4o as target model

| GPT-4o (Source) | | | | Claude-3.5-Haiku (Source) | | | | Gemini-1.5-Flash (Source) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Standard Prompt | CoT | SPP | SPP+ThoT | Standard Prompt | CoT | SPP | SPP+ThoT | Standard Prompt | CoT | SPP | SPP+ThoT |
| 0.72 | 0.76 | 0.82 | 0.82 | 0.9 | 0.78 | 0.88 | 0.78 | 0.78 | 0.82 | 0.74 | 0.82 |
| 0.76 | 0.72 | 0.78 | 0.7 | 0.76 | 0.76 | 0.82 | 0.82 | 0.76 | 0.78 | 0.64 | 0.82 |
| 0.72 | 0.76 | 0.76 | 0.82 | 0.76 | 0.78 | 0.82 | 0.82 | 0.76 | 0.7 | 0.82 | 0.76 |
| 0.74 | 0.75 | 0.78 | 0.64 | 0.76 | 0.74 | 0.82 | 0.74 | 0.74 | 0.74 | 0.76 | 0.82 |
| 0.76 | 0.76 | 0.74 | 0.76 | 0.74 | 0.76 | 0.82 | 0.82 | 0.76 | 0.74 | 0.66 | 0.84 |
| 0.76 | 0.78 | 0.82 | 0.74 | 0.84 | 0.8 | 0.9 | 0.82 | 0.78 | 0.78 | 0.88 | 0.82 |
| 0.76 | 0.76 | 0.76 | 0.74 | 0.84 | 0.8 | 0.86 | 0.76 | 0.76 | 0.76 | 0.72 | 0.8 |
| 0.74 | 0.74 | 0.74 | 0.76 | 0.74 | 0.72 | 0.82 | 0.76 | 0.76 | 0.74 | 0.74 | 0.74 |
| 0.76 | 0.72 | 0.8 | 0.82 | 0.8 | 0.74 | 0.82 | 0.82 | 0.82 | 0.76 | 0.76 | 0.84 |
| 0.76 | 0.74 | 0.74 | 0.7 | 0.76 | 0.74 | 0.74 | 0.76 | 0.76 | 0.76 | 0.2 | 0.88 |
| 0.78 | 0.8 | 0.78 | 0.82 | 0.76 | 0.82 | 0.82 | 0.82 | 0.78 | 0.76 |  | 0.86 |
| 0.72 | 0.76 | 0.8 | 0.82 | 0.82 | 0.78 | 0.84 | 0.82 | 0.76 | 0.74 | 0.82 | 0.82 |
| 0.8 | 0.82 | 0.82 | 0.78 | 0.82 | 0.82 | 0.84 | 0.9 | 0.82 | 0.82 | 0.8 | 0.82 |
| 0.76 | 0.74 | 0.74 | 0.76 | 0.78 | 0.76 | 0.78 | 0.82 | 0.76 | 0.76 | 0.74 | 0.82 |
| 0.74 | 0.76 | 0.76 | 0.76 | 0.76 | 0.76 | 0.78 | 0.76 | 0.76 | 0.72 | 0.74 | 0.74 |
| 0.82 | 0.82 | 0.82 | 0.82 | 0.82 | 0.84 | 0.96 | 0.84 | 0.82 | 0.82 | 0.82 | 0.84 |
| 0.64 | 0.74 | 0.76 | 0.74 | 0.76 | 0.82 | 0.84 | 0.82 | 0.74 | 0.66 | 0.74 | 0.82 |
| 0.84 | 0.76 | 0.76 | 0.8 | 0.78 | 0.74 | 0.84 | 0.86 | 0.7 | 0.72 | 0.79 | 0.86 |
| 0.76 | 0.8 | 0.78 | 0.82 | 0.84 | 0.82 | 0.84 | 0.86 | 0.76 | 0.76 | 0.82 | 0.82 |
| 0.72 | 0.74 | 0.82 | 0.82 | 0.76 | 0.74 | 0.82 | 0.76 | 0.8 | 0.76 | 0.82 | 0.82 |
| 0.72 | 0.82 | 0.74 | 0.76 | 0.76 | 0.76 | 0.82 | 0.86 | 0.76 | 0.76 |  | 0.82 |
| 0.76 | 0.82 | 0.78 | 0.82 | 0.84 | 0.82 | 0.82 | 0.82 | 0.74 | 0.76 | 0.5 | 0.88 |
| 0.74 | 0.78 | 0.76 | 0.74 | 0.8 | 0.8 | 0.74 | 0.76 | 0.76 | 0.74 | 0.76 | 0.82 |
| 0.74 | 0.76 | 0.74 | 0.82 | 0.76 | 0.82 | 0.82 | 0.82 | 0.74 | 0.78 | 0.76 | 0.92 |
| 0.76 | 0.74 | 0.82 | 0.76 | 0.86 | 0.76 | 0.8 | 0.8 | 0.82 | 0.8 | 0.72 | 0.78 |
| 0.82 | 0.82 | 0.82 | 0.86 | 0.92 | 0.92 | 0.9 | 0.94 | 0.82 | 0.86 | 0.82 | 0.82 |
| 0.76 | 0.78 | 0.82 | 0.78 | 0.74 | 0.76 | 0.82 | 0.82 | 0.76 | 0.68 | 0.8 | 0.84 |
| 0.74 | 0.76 | 0.82 | 0.7 | 0.74 | 0.7 | 0.82 | 0.7 | 0.78 | 0.65 |  | 0.82 |
| 0.82 | 0.76 | 0.84 | 0.84 | 0.84 | 0.88 | 0.84 | 0.88 | 0.8 | 0.78 | 0.8 | 0.82 |
| 0.72 | 0.74 | 0.74 | 0.74 | 0.76 | 0.74 |  | 0.82 | 0.74 | 0.74 | 0.82 | 0.82 |
| 0.76 | 0.74 | 0.84 | 0.82 | 0.8 | 0.76 | 0.82 | 0.78 | 0.78 | 0.76 |  | 0.82 |
| 0.8 | 0.74 | 0.76 | 0.74 | 0.82 | 0.76 | 0.82 | 0.88 | 0.76 | 0.74 | 0.76 | 0.82 |
| 0.74 | 0.75 | 0.76 | 0.76 | 0.76 | 0.76 | 0.76 | 0.82 | 0.76 | 0.78 |  | 0.82 |
| 0.74 | 0.76 | 0.74 | 0.78 | 0.76 | 0.78 | 0.74 | 0.82 | 0.74 | 0.7 | 0.82 | 0.82 |
| 0.74 | 0.76 | 0.82 | 0.82 | 0.78 | 0.76 | 0.82 | 0.82 | 0.78 | 0.76 | 0.7 | 0.82 |
| 0.76 | 0.7 | 0.74 | 0.76 | 0.78 | 0.76 | 0.76 | 0.76 | 0.78 | 0.7 |  | 0.76 |
| 0.74 | 0.74 | 0.76 | 0.82 | 0.82 | 0.76 | 0.82 | 0.84 | 0.74 | 0.7 | 0.78 | 0.76 |
| 0.76 | 0.8 | 0.82 | 0.76 | 0.76 | 0.76 | 0.82 | 0.82 | 0.76 | 0.74 | 0.74 | 0.82 |
| 0.66 | 0.76 | 0.78 | 0.82 | 0.74 | 0.82 | 0.84 | 0.82 | 0.76 | 0.76 | 0.76 | 0.82 |
| 0.82 | 0.74 | 0.82 | 0.76 | 0.82 | 0.82 | 0.82 | 0.9 | 0.76 | 0.76 | 0.83 | 0.82 |
| 0.76 | 0.76 | 0.76 | 0.82 | 0.8 | 0.82 | 0.82 | 0.82 | 0.76 | 0.78 | 0.82 | 0.82 |
| 0.76 | 0.74 | 0.76 | 0.76 | 0.82 | 0.82 | 0.84 | 0.84 | 0.74 | 0.78 | 0.72 | 0.82 |
| 0.76 | 0.72 | 0.8 | 0.82 | 0.8 | 0.78 | 0.94 | 0.82 | 0.66 | 0.76 | 0.74 | 0.82 |
| 0.72 | 0.66 | 0.8 | 0.7 | 0.76 | 0.68 | 0.82 | 0.66 | 0.76 | 0.76 | 0.74 | 0.82 |
| 0.76 | 0.82 | 0.84 | 0.84 | 0.82 | 0.82 |  | 0.88 | 0.76 | 0.72 | 0.82 | 0.86 |
| 0.82 | 0.82 | 0.8 | 0.82 | 0.86 | 0.84 |  | 0.86 | 0.82 | 0.82 | 0.82 | 0.84 |
| 0.76 | 0.76 | 0.76 | 0.82 | 0.76 | 0.78 | 0.82 | 0.86 | 0.76 | 0.76 | 0.76 | 0.76 |
| 0.76 | 0.76 | 0.76 | 0.76 | 0.82 | 0.82 | 0.84 | 0.86 | 0.82 | 0.72 | 0.84 | 0.8 |
| 0.76 | 0.82 | 0.82 | 0.82 | 0.82 | 0.82 | 0.82 | 0.82 | 0.76 | 0.82 | 0.82 | 0.84 |
| 0.76 | 0.76 | 0.76 | 0.76 | 0.82 | 0.82 | 0.82 | 0.84 | 0.82 | 0.82 | 0.82 | 0.84 |
| 0.76 | 0.82 | 0.76 | 0.78 | 0.82 | 0.76 | 0.82 | 0.82 | 0.76 | 0.76 | 0.82 | 0.82 |
| 0.84 | 0.78 | 0.84 | 0.82 | 0.88 | 0.86 | 0.94 | 0.94 | 0.82 | 0.82 | 0.86 | 0.82 |
| 0.82 | 0.74 | 0.74 | 0.76 | 0.84 | 0.76 | 0.76 | 0.76 | 0.76 | 0.76 |  | 0.82 |
| 0.78 | 0.76 | 0.82 | 0.74 | 0.82 | 0.82 | 0.84 | 0.86 | 0.76 | 0.72 | 0.82 | 0.86 |
| 0.84 | 0.84 | 0.82 | 0.82 | 0.82 | 0.82 | 0.82 | 0.86 | 0.84 | 0.82 | 0.82 | 0.84 |
| 0.76 | 0.74 | 0.82 | 0.82 | 0.76 | 0.76 | 0.84 | 0.82 | 0.74 | 0.8 | 0.72 | 0.82 |
| 0.82 | 0.78 | 0.82 | 0.82 | 0.82 | 0.82 | 0.84 | 0.84 | 0.8 | 0.82 | 0.82 | 0.84 |
| 0.76 | 0.7 | 0.7 | 0.7 | 0.78 | 0.7 | 0.76 | 0.76 | 0.72 | 0.68 | 0.72 | 0.84 |
| 0.74 | 0.76 | 0.78 | 0.82 | 0.82 | 0.82 | 0.84 | 0.82 | 0.82 | 0.74 | 0.82 | 0.82 |
| 0.76 | 0.76 | 0.76 | 0.76 | 0.76 | 0.82 | 0.8 | 0.82 | 0.76 | 0.76 | 0.82 | 0.8 |
| 0.74 | 0.75 | 0.76 | 0.65 | 0.74 | 0.72 | 0.74 | 0.8 | 0.7 | 0.83 | 0.72 | 0.76 |
| 0.76 | 0.72 | 0.78 | 0.74 | 0.74 | 0.72 | 0.78 | 0.76 | 0.78 | 0.72 | 0.82 | 0.84 |
| 0.8 | 0.78 | 0.74 | 0.78 | 0.82 | 0.82 | 0.86 | 0.86 | 0.76 | 0.76 | 0.76 | 0.82 |
| 0.8 | 0.76 | 0.84 | 0.78 | 0.82 | 0.76 | 0.84 | 0.9 | 0.76 | 0.78 | 0.82 | 0.86 |
| 0.72 | 0.76 | 0.78 | 0.72 | 0.84 | 0.78 | 0.82 | 0.82 | 0.76 | 0.78 |  | 0.82 |
| 0.7 | 0.7 | 0.7 | 0.7 | 0.74 | 0.76 | 0.76 | 0.76 | 0.68 | 0.66 | 0.74 | 0.66 |
| 0.76 | 0.72 | 0.76 | 0.76 | 0.76 | 0.74 | 0.74 | 0.76 | 0.76 | 0.72 | 0.76 | 0.82 |
| 0.76 | 0.76 | 0.82 | 0.82 | 0.82 | 0.82 | 0.82 | 0.84 | 0.74 | 0.76 | 0.76 | 0.88 |
| 0.76 | 0.66 | 0.7 | 0.76 | 0.82 | 0.76 | 0.76 | 0.82 | 0.78 | 0.76 | 0.76 | 0.76 |
| 0.76 | 0.72 | 0.82 | 0.74 | 0.84 | 0.8 | 0.82 | 0.86 | 0.82 | 0.74 | 0.82 | 0.82 |
| 0.78 | 0.7 | 0.78 | 0.76 | 0.84 | 0.84 | 0.78 | 0.84 | 0.76 | 0.78 | 0.78 | 0.84 |

## Appendix 1: Continues

| GPT-4o (Source) | | | | Claude-3.5-Haiku (Source) | | | | Gemini-1.5-Flash (Source) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Standard Prompt | CoT | SPP | SPP+ThoT | Standard Prompt | CoT | SPP | SPP+ThoT | Standard Prompt | CoT | SPP | SPP+ThoT |
| 0.78 | 0.76 | 0.82 | 0.76 | 0.76 | 0.76 | 0.78 | 0.82 | 0.76 | 0.72 | 0.82 | 0.82 |
| 0.76 | 0.74 | 0.68 | 0.82 | 0.84 | 0.82 | 0.82 | 0.9 | 0.76 | 0.82 | 0.82 | 0.82 |
| 0.72 | 0.74 | 0.74 | 0.72 | 0.76 | 0.76 | 0.76 | 0.76 | 0.76 | 0.74 | 0.74 | 0.76 |
| 0.78 | 0.76 | 0.84 | 0.8 | 0.82 | 0.78 | 0.86 | 0.86 | 0.76 | 0.76 | 0.82 | 0.82 |
| 0.72 | 0.74 | 0.78 | 0.78 | 0.76 | 0.76 | 0.74 | 0.74 | 0.74 | 0.74 |  | 0.82 |
| 0.76 | 0.86 | 0.82 | 0.76 | 0.84 | 0.76 |  | 0.79 | 0.76 | 0.74 | 0.7 | 0.86 |
| 0.82 | 0.76 | 0.74 | 0.82 | 0.84 | 0.8 | 0.84 | 0.9 | 0.78 | 0.72 |  | 0.82 |
| 0.72 | 0.76 | 0.8 | 0.74 | 0.78 | 0.76 | 0.86 | 0.82 | 0.76 | 0.76 | 0.82 | 0.84 |
| 0.66 | 0.82 | 0.76 | 0.76 | 0.82 | 0.82 | 0.8 | 0.88 | 0.78 | 0.72 | 0.76 | 0.8 |
| 0.74 | 0.76 | 0.76 | 0.76 | 0.8 | 0.82 | 0.82 | 0.84 | 0.72 | 0.82 |  | 0.82 |
| 0.7 | 0.76 | 0.78 | 0.82 | 0.66 | 0.82 | 0.84 | 0.84 | 0.64 | 0.74 | 0.82 | 0.82 |
| 0.7 | 0.76 | 0.72 | 0.76 | 0.82 | 0.84 | 0.76 | 0.82 | 0.74 | 0.74 |  | 0.82 |
| 0.76 | 0.66 | 0.74 | 0.76 | 0.76 | 0.76 | 0.74 | 0.76 | 0.76 | 0.76 | 0.74 | 0.76 |
| 0.75 | 0.76 | 0.8 | 0.82 | 0.78 | 0.78 | 0.82 | 0.82 | 0.74 | 0.82 | 0.84 | 0.82 |
| 0.74 | 0.74 | 0.82 | 0.78 | 0.76 | 0.82 | 0.8 | 0.78 | 0.74 | 0.76 | 0.76 | 0.82 |
| 0.76 | 0.76 | 0.74 | 0.8 | 0.82 | 0.78 | 0.82 | 0.8 | 0.76 | 0.76 |  | 0.82 |
| 0.76 | 0.76 | 0.76 | 0.78 | 0.82 | 0.8 | 0.84 | 0.82 | 0.76 | 0.74 | 0.7 | 0.8 |
| 0.82 | 0.82 | 0.82 | 0.8 | 0.9 | 0.76 | 0.8 | 0.82 | 0.82 | 0.82 | 0.82 | 0.92 |
| 0.78 | 0.76 | 0.82 | 0.82 | 0.76 | 0.72 | 0.82 | 0.86 | 0.76 | 0.76 |  | 0.82 |
| 0.76 | 0.82 | 0.82 | 0.82 | 0.84 | 0.82 | 0.84 | 0.86 | 0.82 | 0.82 | 0.82 | 0.88 |
| 0.76 | 0.76 | 0.82 | 0.82 | 0.74 | 0.72 | 0.8 | 0.82 | 0.74 | 0.76 | 0.82 | 0.76 |
| 0.76 | 0.76 | 0.78 | 0.76 | 0.82 | 0.76 | 0.82 | 0.76 | 0.76 | 0.74 | 0.78 | 0.82 |
| 0.74 | 0.72 | 0.82 | 0.82 | 0.74 | 0.76 | 0.86 | 0.82 | 0.78 | 0.74 |  | 0.84 |
| 0.76 | 0.74 | 0.82 | 0.82 | 0.76 | 0.76 | 0.82 | 0.86 | 0.78 | 0.72 | 0.74 | 0.82 |
| 0.78 | 0.82 | 0.74 | 0.82 | 0.82 | 0.86 | 0.82 | 0.84 | 0.82 | 0.74 |  | 0.82 |
| 0.72 | 0.78 | 0.74 | 0.76 | 0.76 | 0.76 | 0.82 | 0.82 | 0.76 | 0.72 |  | 0.82 |
| 0.76 | 0.76 | 0.7 | 0.82 | 0.76 | 0.72 | 0.8 | 0.86 | 0.76 | 0.72 |  | 0.82 |
| 0.76 | 0.82 | 0.76 | 0.76 | 0.82 | 0.82 | 0.88 | 0.86 | 0.78 | 0.76 | 0.82 | 0.82 |
| 0.76 | 0.76 | 0.74 | 0.82 | 0.8 | 0.84 | 0.84 | 0.8 | 0.74 | 0.66 |  | 0.82 |

## Appendix 2: Evaluation results of Claude-3.5-Haiku as target model

| GPT-4o (Source) | | | | Claude-3.5-Haiku (Source) | | | | Gemini-1.5-Flash (Source) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Standard Prompt | CoT | SPP | SPP+ThoT | Standard Prompt | CoT | SPP | SPP+ThoT | Standard Prompt | CoT | SPP | SPP+ThoT |
| 0.92 | 0.86 | 0.92 | 0.9 | 0.9 | 0.74 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.88 |
| 0.82 | 0.88 | 0.86 | 0.96 | 0.74 | 0.82 | 0.78 | 0.54 | 0.82 | 0.88 | 0.82 | 0.82 |
| 0.82 | 0.76 | 0.94 | 0.82 | 0.82 | 0.74 | 0.74 | 0.82 | 0.82 | 0.66 | 0.82 | 0.82 |
| 0.82 | 0.92 | 0.88 | 0.82 | 0.7 | 0.82 | 0.82 | 0.74 | 0.86 | 0.92 | 0.86 | 0.9 |
| 0.9 | 0.9 | 0.78 | 0.82 | 0.74 | 0.86 | 0.74 | 0.74 | 0.76 | 0.84 | 0.78 | 0.9 |
| 0.9 | 0.9 | 0.94 | 0.9 | 0.9 | 0.9 | 0.82 | 0.9 | 0.9 | 0.76 | 0.9 | 0.9 |
| 0.9 | 0.9 | 0.94 | 0.9 | 0.74 | 0.78 | 0.8 | 0.74 | 0.74 | 0.82 | 0.68 | 0.9 |
| 0.9 | 0.76 | 0.74 | 0.7 | 0.82 | 0.82 | 0.74 | 0.7 | 0.74 | 0.82 | 0.74 | 0.9 |
| 0.94 | 0.9 | 0.9 | 0.92 | 0.82 | 0.86 | 0.82 | 0.82 | 0.9 | 0.9 | 0.9 | 0.92 |
| 0.78 | 0.82 | 0.86 | 0.82 | 0.72 | 0.78 | 0.82 | 0.66 | 0.74 | 0.82 |  | 0.92 |
| 0.9 | 0.96 | 0.92 | 0.92 | 0.9 | 0.92 | 0.9 | 0.9 | 0.9 | 0.9 |  | 0.94 |
| 0.86 | 0.82 | 0.86 | 0.9 | 0.74 | 0.82 | 0.82 | 0.82 | 0.82 | 0.76 | 0.84 | 0.86 |
| 0.94 | 0.94 | 0.94 | 0.9 | 0.9 | 0.92 | 0.74 | 0.82 | 0.9 | 0.9 | 0.8 | 0.94 |
| 0.9 | 0.9 | 0.9 | 0.9 | 0.68 | 0.78 | 0.78 | 0.76 | 0.82 | 0.9 | 0.74 | 0.94 |
| 0.9 | 0.9 | 0.76 | 0.76 | 0.86 | 0.86 | 0.74 | 0.82 | 0.9 | 0.9 | 0.68 | 0.82 |
| 0.96 | 0.94 | 0.94 | 0.94 | 0.86 | 0.9 | 0.86 | 0.9 | 0.94 | 0.94 | 0.94 | 0.94 |
| 0.82 | 0.92 | 0.86 | 0.86 | 0.74 | 0.74 | 0.74 | 0.74 | 0.76 | 0.68 | 0.74 | 0.76 |
| 0.82 | 0.74 | 0.86 | 0.86 | 0.82 | 0.74 | 0.74 | 0.74 | 0.64 | 0.72 | 0.82 | 0.86 |
| 0.9 | 0.9 | 0.92 | 0.94 | 0.86 | 0.79 | 0.9 | 0.9 | 0.8 | 0.76 | 0.86 | 0.82 |
| 0.94 | 0.88 | 0.84 | 0.9 | 0.82 | 0.8 | 0.9 | 0.74 | 0.9 | 0.78 | 0.9 | 0.9 |
| 0.82 | 0.82 | 0.9 | 0.94 | 0.74 | 0.74 | 0.74 | 0.8 | 0.74 | 0.78 |  | 0.82 |
| 0.92 | 0.9 | 0.86 | 0.9 | 0.9 | 0.8 | 0.78 | 0.9 | 0.86 | 0.82 | 0.56 | 0.9 |
| 0.86 | 0.9 | 0.74 | 0.74 | 0.74 | 0.86 | 0.74 | 0.86 | 0.86 | 0.94 | 0.74 | 0.9 |
| 0.9 | 0.94 | 0.94 | 0.94 | 0.86 | 0.9 | 0.9 | 0.9 | 0.86 | 0.9 | 0.9 | 0.94 |
| 0.82 | 0.84 | 0.78 | 0.84 | 0.79 | 0.8 | 0.76 | 0.76 | 0.82 | 0.78 | 0.76 | 0.86 |
| 0.94 | 0.94 | 0.92 | 0.94 | 0.82 | 0.86 | 0.86 | 0.92 | 0.92 | 0.92 | 0.92 | 0.94 |
| 0.9 | 0.9 | 0.86 | 0.9 | 0.82 | 0.9 | 0.82 | 0.82 | 0.9 | 0.9 | 0.9 | 0.9 |
| 0.9 | 0.9 | 0.82 | 0.78 | 0.82 | 0.86 | 0.9 | 0.88 | 0.9 | 0.9 |  | 0.92 |
| 0.94 | 0.94 | 0.92 | 0.94 | 0.86 | 0.82 | 0.82 | 0.78 | 0.86 | 0.74 | 0.92 | 0.84 |
| 0.9 | 0.9 |  | 0.92 | 0.72 | 0.82 | 0.86 | 0.74 | 0.9 | 0.82 | 0.92 | 0.94 |
| 0.9 | 0.82 | 0.82 | 0.9 | 0.74 | 0.82 | 0.9 | 0.88 | 0.9 | 0.86 |  | 0.9 |
| 0.9 | 0.76 | 0.9 | 0.92 | 0.74 | 0.76 | 0.74 | 0.74 | 0.74 | 0.78 | 0.82 | 0.9 |
| 0.9 | 0.86 | 0.86 | 0.9 | 0.82 | 0.82 | 0.82 | 0.74 | 0.88 | 0.9 |  | 0.9 |
| 0.86 | 0.9 | 0.9 | 0.94 | 0.7 | 0.82 | 0.9 | 0.9 | 0.86 | 0.82 | 0.9 | 0.94 |
| 0.82 | 0.74 | 0.79 | 0.88 | 0.74 | 0.74 | 0.86 | 0.94 | 0.82 | 0.82 | 0.66 | 0.92 |
| 0.8 | 0.9 | 0.86 | 0.79 | 0.74 | 0.74 | 0.68 | 0.74 | 0.76 | 0.84 |  | 0.82 |
| 0.82 | 0.8 | 0.86 | 0.88 | 0.82 | 0.76 | 0.82 | 0.74 | 0.82 | 0.8 | 0.82 | 0.82 |
| 0.9 | 0.88 | 0.86 | 0.82 | 0.74 | 0.82 | 0.8 | 0.66 | 0.9 | 0.9 | 0.74 | 0.9 |
| 0.82 | 0.94 | 0.82 | 0.86 | 0.82 | 0.68 | 0.74 | 0.86 | 0.82 | 0.76 | 0.82 | 0.76 |
| 0.9 | 0.92 | 0.82 | 0.94 | 0.9 | 0.86 | 0.78 | 0.74 | 0.74 | 0.86 | 0.82 | 0.86 |
| 0.9 | 0.88 | 0.88 | 0.9 | 0.86 | 0.8 | 0.74 | 0.9 | 0.9 | 0.86 |  | 0.9 |
| 0.86 | 0.86 | 0.9 | 0.86 | 0.78 | 0.74 | 0.74 | 0.74 | 0.82 | 0.82 | 0.79 | 0.86 |
| 0.9 | 0.82 | 0.9 | 0.92 | 0.76 | 0.86 | 0.9 | 0.8 | 0.76 | 0.9 | 0.76 | 0.94 |
| 0.82 | 0.7 | 0.76 | 0.66 | 0.66 | 0.76 | 0.74 | 0.7 | 0.74 | 0.9 |  | 0.9 |
| 0.9 | 0.94 |  | 0.94 | 0.82 | 0.9 | 0.92 | 0.86 | 0.74 | 0.82 | 0.86 | 0.94 |
| 0.96 | 0.96 |  | 0.96 | 0.88 | 0.88 | 0.82 | 0.86 | 0.94 | 0.88 | 0.94 | 0.94 |
| 0.86 | 0.82 | 0.86 | 0.86 | 0.76 | 0.74 | 0.82 | 0.82 | 0.82 | 0.74 | 0.76 | 0.76 |
| 0.88 | 0.86 | 0.94 | 0.94 | 0.76 | 0.74 | 0.8 | 0.8 | 0.92 | 0.82 | 0.82 | 0.86 |
| 0.94 | 0.82 | 0.86 | 0.94 | 0.74 | 0.82 | 0.79 | 0.82 | 0.74 | 0.9 | 0.92 | 0.94 |
| 0.94 | 0.86 | 0.82 | 0.86 | 0.74 | 0.74 | 0.78 | 0.76 | 0.86 | 0.86 | 0.78 | 0.86 |
| 0.9 | 0.9 | 0.76 | 0.82 | 0.74 | 0.9 | 0.9 | 0.74 | 0.9 | 0.76 | 0.9 | 0.9 |
| 0.94 | 0.96 | 0.96 | 0.96 | 0.9 | 0.9 | 0.92 | 0.94 | 0.96 | 0.9 | 0.94 | 0.94 |
| 0.9 | 0.86 | 0.74 | 0.9 | 0.82 | 0.86 | 0.64 | 0.74 | 0.9 | 0.9 |  | 0.74 |
| 0.86 | 0.88 | 0.94 | 0.94 | 0.74 | 0.74 | 0.82 | 0.74 | 0.74 | 0.76 | 0.8 | 0.8 |
| 0.94 | 0.94 | 0.92 | 0.96 | 0.94 | 0.94 | 0.9 | 0.88 | 0.96 | 0.9 | 0.86 | 0.94 |
| 0.88 | 0.86 | 0.86 | 0.86 | 0.82 | 0.82 | 0.82 | 0.82 | 0.9 | 0.82 |  | 0.82 |
| 0.94 | 0.94 | 0.94 | 0.94 | 0.88 | 0.9 | 0.9 | 0.94 | 0.9 | 0.9 | 0.94 | 0.94 |
| 0.74 | 0.82 | 0.9 | 0.86 | 0.64 | 0.76 | 0.62 | 0.74 | 0.74 | 0.74 | 0.82 | 0.9 |
| 0.94 | 0.86 | 0.94 | 0.9 | 0.74 | 0.74 | 0.82 | 0.86 | 0.86 | 0.74 | 0.8 | 0.88 |
| 0.88 | 0.82 | 0.82 | 0.86 | 0.74 | 0.74 | 0.64 | 0.82 | 0.82 | 0.74 | 0.78 | 0.86 |
| 0.86 | 0.86 | 0.74 | 0.82 | 0.8 | 0.84 | 0.7 | 0.86 | 0.92 | 0.92 | 0.54 | 0.9 |
| 0.82 | 0.82 | 0.74 | 0.9 | 0.86 | 0.9 | 0.78 | 0.74 | 0.74 | 0.82 | 0.74 | 0.9 |
| 0.94 | 0.88 | 0.82 | 0.96 | 0.72 | 0.82 | 0.74 | 0.74 | 0.74 | 0.76 | 0.74 | 0.76 |
| 0.9 | 0.88 | 0.86 | 0.94 | 0.78 | 0.66 | 0.78 | 0.82 | 0.74 | 0.82 | 0.82 | 0.9 |
| 0.88 | 0.88 | 0.9 | 0.88 | 0.8 | 0.74 | 0.74 | 0.74 | 0.82 | 0.86 |  | 0.88 |
| 0.86 | 0.84 | 0.78 | 0.74 | 0.66 | 0.72 | 0.74 | 0.66 | 0.72 | 0.78 | 0.78 | 0.7 |
| 0.82 | 0.66 | 0.7 | 0.7 | 0.66 | 0.82 | 0.74 | 0.74 | 0.86 | 0.78 | 0.68 | 0.82 |
| 0.86 | 0.78 | 0.82 | 0.86 | 0.74 | 0.64 | 0.82 | 0.76 | 0.74 | 0.74 | 0.82 | 0.8 |
| 0.86 | 0.9 | 0.88 | 0.86 | 0.82 | 0.66 | 0.74 | 0.74 | 0.86 | 0.86 | 0.86 | 0.88 |
| 0.9 | 0.9 | 0.9 | 0.86 | 0.8 | 0.78 | 0.82 | 0.74 | 0.9 | 0.86 | 0.84 | 0.86 |
| 0.86 | 0.86 | 0.86 | 0.86 | 0.86 | 0.74 | 0.8 | 0.82 | 0.8 | 0.9 | 0.82 | 0.86 |
| 0.78 | 0.88 | 0.78 | 0.74 | 0.68 | 0.74 | 0.74 | 0.66 | 0.78 | 0.74 | 0.84 | 0.9 |
| 0.86 | 0.96 | 0.9 | 0.96 | 0.74 | 0.74 | 0.7 | 0.9 | 0.74 | 0.82 | 0.82 | 0.94 |
| 0.82 | 0.86 | 0.76 | 0.86 | 0.66 | 0.86 | 0.82 | 0.76 | 0.86 | 0.9 | 0.76 | 0.74 |

## Appendix 2: Continues

| GPT-4o | | | | Claude-3.5-Haiku | | | | Gemini-1.5-Flash | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Standard Prompt | CoT | SPP | SPP+ThoT | Standard Prompt | CoT | SPP | SPP+ThoT | Standard Prompt | CoT | SPP | SPP+ThoT |
| 0.96 | 0.82 | 0.9 | 0.92 | 0.76 | 0.78 | 0.9 | 0.74 | 0.9 | 0.78 | 0.9 | 0.9 |
| 0.9 | 0.82 | 0.74 | 0.74 | 0.78 | 0.88 | 0.74 | 0.74 | 0.88 | 0.82 | | 0.78 |
| 0.9 | 0.82 | | 0.9 | 0.74 | 0.9 | 0.86 | 0.82 | 0.9 | 0.9 | 0.8 | 0.88 |
| 0.92 | 0.82 | 0.92 | 0.9 | 0.74 | 0.78 | 0.74 | 0.82 | 0.9 | 0.82 | | 0.92 |
| 0.9 | 0.9 | 0.9 | 0.88 | 0.76 | 0.76 | 0.78 | 0.76 | 0.74 | 0.74 | 0.86 | 0.92 |
| 0.88 | 0.82 | 0.86 | 0.86 | 0.72 | 0.74 | 0.8 | 0.74 | 0.74 | 0.74 | 0.82 | 0.84 |
| 0.9 | 0.86 | 0.92 | 0.86 | 0.74 | 0.64 | 0.82 | 0.82 | 0.74 | 0.9 | | 0.94 |
| 0.82 | 0.86 | 0.9 | 0.96 | 0.74 | 0.74 | 0.74 | 0.74 | 0.82 | 0.76 | 0.86 | 0.78 |
| 0.86 | 0.86 | 0.86 | 0.86 | 0.62 | 0.82 | 0.74 | 0.74 | 0.82 | 0.82 | | 0.9 |
| 0.82 | 0.82 | 0.82 | 0.82 | 0.74 | 0.78 | 0.66 | 0.74 | 0.74 | 0.9 | 0.78 | 0.78 |
| 0.9 | 0.94 | 0.96 | 0.9 | 0.7 | 0.86 | 0.9 | 0.9 | 0.9 | 0.74 | 0.9 | 0.96 |
| 0.82 | 0.9 | 0.9 | 0.9 | 0.74 | 0.82 | 0.84 | 0.86 | 0.86 | 0.9 | 0.82 | 0.92 |
| 0.82 | 0.78 | 0.86 | 0.9 | 0.74 | 0.82 | 0.66 | 0.78 | 0.9 | 0.82 | | 0.88 |
| 0.86 | 0.86 | 0.82 | 0.86 | 0.79 | 0.6 | 0.76 | 0.78 | 0.74 | 0.72 | 0.76 | 0.88 |
| 0.94 | 0.9 | 0.84 | 0.96 | 0.9 | 0.9 | 0.94 | 0.94 | 0.92 | 0.9 | 0.9 | 0.92 |
| 0.9 | 0.76 | 0.8 | 0.9 | 0.82 | 0.9 | 0.84 | 0.74 | 0.74 | 0.9 | | 0.9 |
| 0.94 | 0.96 | 0.9 | 0.94 | 0.88 | 0.9 | 0.9 | 0.9 | 0.94 | 0.88 | 0.86 | 0.94 |
| 0.82 | 0.82 | 0.92 | 0.94 | 0.78 | 0.74 | 0.86 | 0.82 | 0.82 | 0.82 | 0.82 | 0.82 |
| 0.9 | 0.78 | 0.88 | 0.86 | 0.66 | 0.74 | 0.78 | 0.74 | 0.76 | 0.76 | 0.82 | 0.82 |
| 0.74 | 0.76 | 0.92 | 0.9 | 0.74 | 0.72 | 0.8 | 0.74 | 0.82 | 0.82 | | 0.86 |
| 0.82 | 0.86 | 0.82 | 0.9 | 0.74 | 0.82 | 0.74 | 0.86 | 0.86 | 0.82 | 0.8 | 0.9 |
| 0.96 | 0.86 | 0.9 | 0.86 | 0.74 | 0.86 | 0.74 | 0.86 | 0.9 | 0.74 | | 0.86 |
| 0.9 | 0.9 | 0.94 | 0.9 | 0.82 | 0.9 | 0.82 | 0.74 | 0.9 | 0.86 | | 0.82 |
| 0.82 | 0.74 | 0.86 | 0.94 | 0.82 | 0.74 | 0.64 | 0.82 | 0.74 | 0.74 | | 0.8 |
| 0.9 | 0.82 | 0.86 | 0.96 | 0.72 | 0.82 | 0.64 | 0.74 | 0.88 | 0.74 | 0.79 | 0.9 |
| 0.94 | 0.94 | 0.94 | 0.86 | 0.8 | 0.7 | 0.68 | 0.78 | 0.74 | 0.82 | | 0.92 |

## Appendix 3: Evaluation results of Gemini-1.5-Flash as target model

| GPT-4o (Source) | | | | Claude-3.5-Haiku (Source) | | | | Gemini-1.5-Flash (Source) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Standard Prompt | CoT | SPP | SPP+ThoT | Standard Prompt | CoT | SPP | SPP+ThoT | Standard Prompt | CoT | SPP | SPP+ThoT |
| 0.92 | 0.86 | 0.92 | 0.9 | 0.9 | 0.74 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.88 |
| 0.86 | 0.78 | 0.92 | 0.78 | 0.74 | 0.76 | 0.88 | 0.8 | 0.76 | 0.78 | 0.82 | 0.94 |
| 0.78 | 0.78 | 0.94 | 0.98 | 0.88 | 0.66 | 0.72 | 0.64 | 0.74 | 0.78 | 0.78 | 0.96 |
| 0.66 | 0.86 | 0.78 | 0.98 | 0.84 | 0.74 | 0.76 | 0.74 | 0.78 | 0.74 | 0.82 | 0.86 |
| 0.78 | 0.78 | 0.74 | 0.78 | 0.78 | 0.66 | 0.66 | 0.66 | 0.78 | 0.84 | 0.88 | 0.86 |
| 0.86 | 0.72 | 0.78 | 0.84 | 0.84 | 0.88 | 0.74 | 0.8 | 0.76 | 0.78 | 0.88 | 0.92 |
| 0.8 | 0.78 | 0.96 | 0.78 | 0.74 | 0.76 | 0.74 | 0.7 | 0.82 | 0.72 | 0.96 | 0.88 |
| 0.96 | 0.74 | 0.98 | 0.74 | 0.74 | 0.76 | 0.84 | 0.72 | 0.78 | 0.82 | 0.76 | 0.74 |
| 0.86 | 0.76 | 0.72 | 0.74 | 0.78 | 0.66 | 0.7 | 0.66 | 0.74 | 0.76 | 0.7 | 0.78 |
| 0.78 | 0.78 | 0.82 | 0.94 | 0.76 | 0.84 | 0.82 | 0.78 | 0.78 | 0.78 | 0.74 | 0.88 |
| 0.76 | 0.66 | 0.78 | 0.7 | 0.68 | 0.58 | 0.66 | 0.66 | 0.78 | 0.78 | 0.0 | 0.82 |
| 0.86 | 0.94 | 0.78 | 0.94 | 0.78 | 0.84 | 0.78 | 0.98 | 0.84 | 0.82 | 0.0 | 0.96 |
| 0.86 | 0.74 | 0.96 | 0.96 | 0.84 | 0.78 | 0.78 | 0.82 | 0.84 | 0.74 | 0.84 | 0.86 |
| 0.94 | 0.88 | 0.96 | 0.96 | 0.82 | 0.77 | 0.76 | 0.74 | 0.78 | 0.84 | 0.82 | 0.96 |
| 0.88 | 0.82 | 0.86 | 0.76 | 0.66 | 0.74 | 0.84 | 0.74 | 0.86 | 0.76 | 0.82 | 0.84 |
| 0.74 | 0.76 | 0.74 | 0.66 | 0.88 | 0.78 | 0.74 | 0.72 | 0.7 | 0.72 | 0.76 | 0.74 |
| 0.96 | 0.98 | 0.98 | 0.94 | 0.82 | 0.98 | 0.94 | 0.92 | 0.94 | 0.96 | 0.94 | 0.96 |
| 0.82 | 0.86 | 0.8 | 0.96 | 0.74 | 0.76 | 0.78 | 0.76 | 0.86 | 0.7 | 0.78 | 0.86 |
| 0.74 | 0.78 | 0.88 | 0.88 | 0.74 | 0.74 | 0.7 | 0.8 | 0.76 | 0.7 | 0.82 | 0.86 |
| 0.78 | 0.82 | 0.9 | 0.92 | 0.74 | 0.76 | 0.8 | 0.74 | 0.76 | 0.76 | 0.88 | 0.74 |
| 0.84 | 0.82 | 0.88 | 0.74 | 0.7 | 0.76 | 0.86 | 0.74 | 0.84 | 0.82 | 0.82 | 0.86 |
| 0.82 | 0.84 | 0.88 | 0.98 | 0.64 | 0.8 | 0.7 | 0.74 | 0.78 | 0.78 | 0.0 | 0.74 |
| 0.86 | 0.98 | 0.78 | 0.94 | 0.74 | 0.82 | 0.78 | 0.78 | 0.84 | 0.76 | 0.46 | 0.94 |
| 0.86 | 0.76 | 0.68 | 0.74 | 0.82 | 0.78 | 0.76 | 0.66 | 0.9 | 0.86 | 0.78 | 0.98 |
| 0.92 | 0.94 | 0.94 | 0.96 | 0.86 | 0.82 | 0.8 | 0.88 | 0.86 | 0.94 | 0.86 | 0.96 |
| 0.86 | 0.86 | 0.86 | 0.86 | 0.78 | 0.82 | 0.94 | 0.78 | 0.74 | 0.84 | 0.78 | 0.86 |
| 0.98 | 0.98 | 0.9 | 0.96 | 0.92 | 0.94 | 0.94 | 0.94 | 0.94 | 0.92 | 0.92 | 0.98 |
| 0.92 | 0.88 | 0.8 | 0.94 | 0.9 | 0.86 | 0.76 | 0.7 | 0.78 | 0.9 | 0.9 | 0.96 |
| 0.88 | 0.74 | 0.96 | 0.4 | 0.84 | 0.86 | 0.94 | 0.78 | 0.88 | 0.38 | 0.0 | 0.96 |
| 0.96 | 0.96 | 0.98 | 0.98 | 0.88 | 0.9 | 0.96 | 0.98 | 0.9 | 0.82 | 0.88 | 0.94 |
| 0.86 | 0.84 | 0.0 | 0.94 | 0.68 | 0.76 | 0.78 | 0.58 | 0.84 | 0.78 | 0.86 | 0.88 |
| 0.78 | 0.74 | 0.78 | 0.82 | 0.76 | 0.74 | 0.96 | 0.78 | 0.82 | 0.74 | 0.0 | 0.78 |
| 0.86 | 0.78 | 0.78 | 0.88 | 0.84 | 0.78 | 0.74 | 0.7 | 0.74 | 0.74 | 0.88 | 0.86 |
| 0.66 | 0.7 | 0.86 | 0.78 | 0.7 | 0.74 | 0.86 | 0.7 | 0.46 | 0.84 | 0.0 | 0.8 |
| 0.86 | 0.66 | 0.76 | 0.96 | 0.52 | 0.64 | 0.78 | 0.76 | 0.86 | 0.66 | 0.77 | 0.94 |
| 0.84 | 0.7 | 0.74 | 0.78 | 0.7 | 0.7 | 0.94 | 0.94 | 0.84 | 0.66 | 0.7 | 0.92 |
| 0.86 | 0.78 | 0.94 | 0.86 | 0.66 | 0.76 | 0.76 | 0.76 | 0.88 | 0.78 | 0.0 | 0.78 |
| 0.86 | 0.78 | 0.94 | 0.88 | 0.78 | 0.78 | 0.82 | 0.78 | 0.86 | 0.74 | 0.84 | 0.78 |
| 0.88 | 0.7 | 0.94 | 0.74 | 0.7 | 0.86 | 0.82 | 0.6 | 0.77 | 0.84 | 0.74 | 0.86 |
| 0.78 | 0.78 | 0.82 | 0.9 | 0.7 | 0.78 | 0.78 | 0.78 | 0.76 | 0.78 | 0.78 | 0.86 |
| 0.78 | 0.86 | 0.94 | 0.86 | 0.74 | 0.74 | 0.78 | 0.66 | 0.8 | 0.74 | 0.5 | 0.78 |
| 0.94 | 0.96 | 0.86 | 0.88 | 0.78 | 0.74 | 0.7 | 0.74 | 0.78 | 0.76 | 0.0 | 0.84 |
| 0.94 | 0.96 | 0.84 | 0.88 | 0.74 | 0.74 | 0.78 | 0.74 | 0.78 | 0.78 | 0.78 | 0.86 |
| 0.94 | 0.87 | 0.94 | 0.84 | 0.76 | 0.74 | 0.78 | 0.78 | 0.74 | 0.76 | 0.74 | 0.92 |
| 0.86 | 0.66 | 0.86 | 0.7 | 0.8 | 0.7 | 0.9 | 0.86 | 0.98 | 0.86 | 0.0 | 0.88 |
| 0.88 | 0.84 | 0.0 | 0.98 | 0.74 | 0.84 | 0.84 | 0.94 | 0.7 | 0.74 | 0.9 | 0.96 |
| 0.98 | 0.94 | 0.16 | 0.96 | 0.78 | 0.86 | 0.76 | 0.94 | 0.94 | 0.94 | 0.82 | 0.94 |
| 0.86 | 0.76 | 0.88 | 0.86 | 0.82 | 0.74 | 0.78 | 0.78 | 0.74 | 0.76 | 0.78 | 0.84 |
| 0.94 | 0.96 | 0.98 | 0.96 | 0.84 | 0.86 | 0.84 | 0.82 | 0.84 | 0.86 | 0.88 | 0.84 |
| 0.94 | 0.86 | 0.94 | 0.94 | 0.78 | 0.8 | 0.78 | 0.84 | 0.82 | 0.8 | 0.86 | 0.84 |
| 0.9 | 0.96 | 0.94 | 0.88 | 0.74 | 0.74 | 0.78 | 0.78 | 0.98 | 0.84 | 0.78 | 0.82 |
| 0.86 | 0.94 | 0.78 | 0.84 | 0.74 | 0.8 | 0.94 | 0.9 | 0.78 | 0.86 | 0.78 | 0.82 |
| 0.98 | 0.98 | 0.98 | 0.98 | 0.98 | 0.86 | 0.98 | 0.96 | 0.98 | 0.94 | 0.96 | 0.96 |
| 0.88 | 0.78 | 0.86 | 0.94 | 0.84 | 0.84 | 0.7 | 0.76 | 0.92 | 0.78 | 0.0 | 0.84 |
| 0.88 | 0.94 | 0.98 | 0.98 | 0.76 | 0.72 | 0.9 | 0.74 | 0.76 | 0.74 | 0.9 | 0.88 |
| 0.94 | 0.98 | 0.88 | 0.98 | 0.92 | 0.98 | 0.88 | 0.66 | 0.9 | 0.8 | 0.88 | 0.98 |
| 0.82 | 0.78 | 0.96 | 0.98 | 0.76 | 0.8 | 0.76 | 0.96 | 0.84 | 0.78 | 0.0 | 0.86 |
| 0.86 | 0.74 | 0.98 | 0.98 | 0.94 | 0.7 | 0.84 | 0.86 | 0.94 | 0.84 | 0.98 | 0.84 |
| 0.78 | 0.76 | 0.74 | 0.88 | 0.76 | 0.73 | 0.7 | 0.7 | 0.76 | 0.76 | 0.84 | 0.98 |
| 0.94 | 0.94 | 0.94 | 0.94 | 0.74 | 0.66 | 0.84 | 0.84 | 0.94 | 0.86 | 0.78 | 0.84 |
| 0.78 | 0.84 | 0.84 | 0.8 | 0.74 | 0.8 | 0.74 | 0.78 | 0.76 | 0.76 | 0.86 | 0.86 |
| 0.78 | 0.86 | 0.46 | 0.7 | 0.86 | 0.74 | 0.74 | 0.86 | 0.84 | 0.86 | 0.94 | 0.78 |
| 0.86 | 0.74 | 0.74 | 0.76 | 0.78 | 0.78 | 0.7 | 0.66 | 0.88 | 0.82 | 0.82 | 0.86 |
| 0.86 | 0.9 | 0.84 | 0.88 | 0.7 | 0.76 | 0.7 | 0.76 | 0.76 | 0.78 | 0.86 | 0.88 |
| 0.94 | 0.98 | 0.94 | 0.96 | 0.76 | 0.78 | 0.94 | 0.8 | 0.86 | 0.86 | 0.8 | 0.84 |
| 0.84 | 0.74 | 0.84 | 0.86 | 0.74 | 0.66 | 0.7 | 0.74 | 0.74 | 0.74 | 0.0 | 0.78 |
| 0.8 | 0.88 | 0.78 | 0.66 | 0.78 | 0.76 | 0.66 | 0.68 | 0.8 | 0.78 | 0.58 | 0.78 |
| 0.86 | 0.74 | 0.74 | 0.72 | 0.74 | 0.76 | 0.82 | 0.7 | 0.74 | 0.76 | 0.88 | 0.78 |
| 0.94 | 0.94 | 0.88 | 0.9 | 0.78 | 0.78 | 0.9 | 0.88 | 0.78 | 0.7 | 0.88 | 0.94 |
| 0.88 | 0.86 | 0.78 | 0.8 | 0.74 | 0.66 | 0.66 | 0.7 | 0.84 | 0.86 | 0.84 | 0.78 |
| 0.86 | 0.84 | 0.86 | 0.98 | 0.86 | 0.74 | 0.86 | 0.7 | 0.86 | 0.76 | 0.94 | 0.86 |
| 0.86 | 0.86 | 0.88 | 0.98 | 0.86 | 0.78 | 0.88 | 0.78 | 0.78 | 0.78 | 0.86 | 0.88 |
| 0.88 | 0.86 | 0.84 | 0.86 | 0.74 | 0.74 | 0.7 | 0.66 | 0.78 | 0.74 | 0.82 | 0.8 |
| 0.98 | 0.94 | 0.82 | 0.98 | 0.76 | 0.74 | 0.62 | 0.82 | 0.66 | 0.84 | 0.84 | 0.94 |

## Appendix 3: Continues

| GPT-4o (Source) | | | | Claude-3.5-Haiku (Source) | | | | Gemini-1.5-Flash (Source) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Standard Prompt | CoT | SPP | SPP+ThoT | Standard Prompt | CoT | SPP | SPP+ThoT | Standard Prompt | CoT | SPP | SPP+ThoT |
| 0.78 | 0.84 | 0.86 | 0.86 | 0.66 | 0.84 | 0.84 | 0.76 | 0.84 | 0.9 | 0.88 | 0.86 |
| 0.94 | 0.74 | 0.9 | 0.82 | 0.78 | 0.88 | 0.94 | 0.76 | 0.84 | 0.76 | 0.96 | 0.94 |
| 0.88 | 0.88 | 0.84 | 0.84 | 0.76 | 0.84 | 0.78 | 0.84 | 0.78 | 0.76 | 0.0 | 0.78 |
| 0.86 | 0.9 | 0.0 | 0.82 | 0.74 | 0.96 | 0.94 | 0.88 | 0.86 | 0.74 | 0.78 | 0.88 |
| 0.86 | 0.76 | 0.86 | 0.98 | 0.76 | 0.76 | 0.7 | 0.74 | 0.82 | 0.8 | 0.0 | 0.94 |
| 0.86 | 0.78 | 0.94 | 0.88 | 0.74 | 0.86 | 0.74 | 0.66 | 0.74 | 0.86 | 0.78 | 0.88 |
| 0.86 | 0.86 | 0.86 | 0.96 | 0.7 | 0.78 | 0.82 | 0.74 | 0.76 | 0.76 | 0.74 | 0.86 |
| 0.86 | 0.86 | 0.94 | 0.96 | 0.76 | 0.76 | 0.76 | 0.7 | 0.76 | 0.78 | 0.0 | 0.94 |
| 0.76 | 0.9 | 0.86 | 0.96 | 0.76 | 0.76 | 0.7 | 0.7 | 0.76 | 0.76 | 0.86 | 0.84 |
| 0.96 | 0.94 | 0.78 | 0.96 | 0.26 | 0.72 | 0.5 | 0.64 | 0.76 | 0.84 | 0.0 | 0.94 |
| 0.86 | 0.74 | 0.88 | 0.86 | 0.8 | 0.74 | 0.74 | 0.78 | 0.86 | 0.76 | 0.57 | 0.8 |
| 0.78 | 0.94 | 0.88 | 0.96 | 0.76 | 0.74 | 0.76 | 0.78 | 0.96 | 0.84 | 0.94 | 0.86 |
| 0.86 | 0.88 | 0.86 | 0.87 | 0.74 | 0.8 | 0.84 | 0.9 | 0.88 | 0.94 | 0.82 | 0.88 |
| 0.86 | 0.78 | 0.74 | 0.74 | 0.76 | 0.72 | 0.66 | 0.86 | 0.78 | 0.76 | 0.0 | 0.86 |
| 0.92 | 0.96 | 0.96 | 0.94 | 0.94 | 0.74 | 0.78 | 0.66 | 0.86 | 0.78 | 0.78 | 0.86 |
| 0.98 | 0.94 | 0.86 | 0.98 | 0.96 | 0.98 | 0.98 | 0.96 | 0.96 | 0.96 | 0.98 | 0.94 |
| 0.76 | 0.74 | 0.78 | 0.96 | 0.74 | 0.84 | 0.86 | 0.7 | 0.86 | 0.86 | 0.12 | 0.82 |
| 0.96 | 0.98 | 0.94 | 0.98 | 0.86 | 0.96 | 0.94 | 0.94 | 0.94 | 0.98 | 0.9 | 0.98 |
| 0.78 | 0.7 | 0.84 | 0.96 | 0.66 | 0.77 | 0.86 | 0.84 | 0.82 | 0.84 | 0.78 | 0.84 |
| 0.84 | 0.76 | 0.78 | 0.86 | 0.74 | 0.76 | 0.74 | 0.74 | 0.7 | 0.82 | 0.82 | 0.82 |
| 0.74 | 0.78 | 0.94 | 0.9 | 0.76 | 0.7 | 0.82 | 0.84 | 0.74 | 0.78 | 0.0 | 0.96 |
| 0.9 | 0.66 | 0.84 | 0.96 | 0.74 | 0.7 | 0.76 | 0.86 | 0.88 | 0.66 | 0.78 | 0.76 |
| 0.94 | 0.84 | 0.88 | 0.86 | 0.76 | 0.86 | 0.78 | 0.84 | 0.78 | 0.76 | 0.0 | 0.86 |
| 0.84 | 0.78 | 0.94 | 0.9 | 0.78 | 0.84 | 0.82 | 0.74 | 0.86 | 0.74 | 0.0 | 0.86 |
| 0.76 | 0.74 | 0.84 | 0.96 | 0.7 | 0.78 | 0.72 | 0.82 | 0.76 | 0.86 | 0.0 | 0.9 |
| 0.88 | 0.98 | 0.94 | 0.98 | 0.78 | 0.84 | 0.68 | 0.8 | 0.84 | 0.86 | 0.84 | 0.78 |