



Understanding Prompt Design

In the previous chapters, we mentioned the term **prompt** several times while referring to user input in ChatGPT and OpenAI models in general.

This chapter focuses in more depth on the importance of prompt design and engineering as a technique to improve the accuracy of the model. Prompts heavily impact the model's generated output: a well-designed prompt can help guide the model toward generating relevant and accurate output, while a poorly designed prompt can lead to irrelevant or confusing output. Finally, it is also important to incorporate ethical considerations into the prompt to prevent the model from generating harmful content.

In this chapter, we will discuss the following topics:

What is a prompt and why is it important?

Zero-, one-, and few-shot learning – typical of transformers models

Principles of well-defined prompts to obtain relevant and consistent results

Avoiding the risk of hidden bias and taking into account ethical considerations in ChatGPT

By the end of this chapter, you will be able to get high-quality results from your ChatGPT and OpenAI model interactions thanks to proper prompt design.

What is a prompt and why is it important?

In the context of generative AI, a prompt refers to the input the user provides to the generative models. When we talk about ChatGPT and OpenAI models, prompts mainly refer to a piece of text in natural language or – as we saw for Codex models – in programming languages.

Prompts are the only way users can control the output generated by those models. As such, there is no surprise in saying that the quality of the prompts used to engage the AI system in a conversation is key to determining its success.

Well-defined prompts are the recipe for a successful conversation that covers the topics of interest; poorly-defined prompts, on the other hand, not only prevent the conversation from being useful to the user but could also lead to potentially misleading content. As we saw in the previous chapter, ChatGPT still suffers from some limitations, such as the 2021 cut-off of its knowledge base, as well as another caveat typical of generative AI systems in general, such as the phenomenon of hallucination.

Namely, imagine that we want to learn about **Convolutional Neural Networks (CNNs)**. We don't have any knowledge about **Artificial Neural Networks (ANNs)**, so we need to start with a gentle introduction. The idea is that, starting from the first output, we can deep dive into relevant elements.

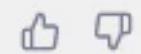
Let's examine the result of a poorly-defined prompt in ChatGPT:

V

tell me about convolutional neural networks



Convolutional Neural Networks (ConvNets or CNNs) are a category of Neural Networks that are specifically designed to process data that has a grid-like topology, such as an image. CNNs are built using a variation of multi-layer perceptrons, designed to require minimal pre-processing, and to operate on input data without fully connected layers, which makes them well-suited for image classification tasks.



A CNN works by passing the input image through multiple layers of operations, each layer extracting higher level features from the input image. The first layer of a CNN, called the convolutional layer, is where the magic happens: it uses a set of learnable filters to scan the image and detect features (edges, curves, textures, etc.) that are relevant for the task. The outputs of these filters are then fed into the next layer, which is typically a pooling layer, responsible for reducing the spatial dimensions of the feature maps, and for suppressing irrelevant information.

Following the pooling layer, there can be multiple layers of fully connected neurons that perform classification. The final layer of a CNN outputs a probability distribution over the classes, which can be used to determine the final class label for an input image.

During training, the parameters (weights and biases) of a CNN are learned using backpropagation and gradient descent, so that the network can learn to associate the input image with the correct class label. The learned filters can be thought of as visual detectors that are specific to the task.

Overall, CNNs have proven to be extremely effective in tasks such as image classification, object detection, semantic segmentation, etc., and they have achieved state-of-the-art performance on several benchmarks.

Figure 4.1 – Example of a poorly-defined prompt

The result is correct and accurate, but it's going through many technical concepts we may have never seen before. What are fully connected layers? What is a pooling layer? This answer assumes we have previous knowledge of ANNs.

Let's try to give more context to ChatGPT to get a result that is more useful to us:

V give me an introduction about convolutional neural network. Assume I have no previous knowledge about artificial neural network. describe the architecture in simple words making examples if possible.

A Convolutional Neural Network (CNN) is a type of Artificial Neural Network that is used for image recognition and processing tasks. It is designed to take advantage of the grid-like structure of an image, where pixels are organized in a 2D grid. Like Edit

Think of a CNN as a machine that is designed to recognize objects in images. It does so by analyzing the image in a series of steps, each step looking for specific features in the image.

For example, the first step might look for edges and curves in the image. The second step could look for repeating textures. And the final step would put all of the features together to recognize an object, like a dog or a cat.

The key to a CNN is its architecture, which consists of multiple layers. Each layer performs a different operation on the image, and the output of one layer is fed as input to the next layer.

Here's a high-level overview of the main components of a CNN:

1. Convolutional Layer: This layer is responsible for detecting local patterns in the image, such as edges, curves, and textures. It does so by using a set of filters that scan the image and create feature maps.
2. Pooling Layer: This layer reduces the size of the feature maps created by the convolutional layer, and helps to reduce the number of parameters in the network.
3. Fully Connected Layer: This layer takes the output of the pooling layer and performs the final classification.

During the training phase, the CNN is presented with many examples of images along with their correct labels (e.g., dog, cat, etc.). The CNN then uses these examples to learn and adjust its parameters, so that it can correctly recognize objects in new images.

In short, a CNN is a deep learning model that is designed to process and analyze images, and it does so by performing multiple operations on the image in a series of layers.

Figure 4.2 – Example of a well-defined prompt

As you can see, the wording in this response is much more likely than the one I was looking for. In the introduction, it doesn't use words such as *multi-layer perceptron* or *fully connected layer*. It also provides, as requested, an example of the process of image recognition. Then, it goes into further details, making sure to explain each step with simple words.

I'm pretty satisfied with this response and, as a user, I could now start asking about the pooling layer in more detail, for example.

Consequently, prompt design and engineering has been gaining more and more traction and it's growing as a discipline itself.

Now, let's focus on how to improve ChatGPT's responses by leveraging its few-shot learning capabilities.

Zero-, one-, and few-shot learning – typical of transformers models

In the previous chapters, we mentioned how OpenAI models, and hence also ChatGPT, come in a pre-trained format. They have been trained on a huge amount of data and have had their (billions of) parameters configured accordingly.

However, this doesn't mean that those models can't learn anymore. In [Chapter 2](#), we saw that one way to customize an OpenAI model and make it more capable of addressing specific tasks is by **fine-tuning**.

Definition

Fine-tuning is the process of adapting a pre-trained model to a new task. In fine-tuning, the parameters of the pre-trained model are altered, either by adjusting the existing parameters or by adding new parameters so that they fit the data for the new task. This is done by training the model on a smaller labeled dataset that is specific to the new task. The key idea behind fine-tuning is to leverage the knowledge learned from the pre-trained model and fine-tune it to the new task, rather than training a model from scratch.

Fine-tuning is a proper training process that requires a training dataset, compute power, and some training time (depending on the amount of data and compute instances).

That is why it is worth testing another method for our model to become more skilled in specific tasks: **shot learning**.

The idea is to let the model learn from simple examples rather than the entire dataset. Those examples are samples of the way we would like the model to respond so that the model not only learns the content but also the format, style, and taxonomy to use in its response.

Furthermore, shot learning occurs directly via the prompt (as we will see in the following scenarios), so the whole experience is less time-consuming and easier to perform.

The number of examples provided determines the level of shot learning we are referring to. In other words, we refer to zero-shot if no example is provided, one-shot if one example is provided, and few-shot if more than 2-3 examples are provided.

Let's focus on each of those scenarios:

Zero-shot learning. In this type of learning, the model is asked to perform a task for which it has not seen any training examples. The model must rely on prior knowledge or general information about the task to complete the task. For example, a zero-shot learning approach could be that of asking the model to generate a description, as defined in my prompt:

V

describe in 2 sentences a landscape with snow and mountains as the introduction of a novel.



The mountains loomed majestically in the distance, their snow-capped peaks piercing the blue sky. The landscape was blanketed in a layer of pristine white snow, stretching as far as the eye could see.



Figure 4.3 – Example of zero-shot learning

One-shot learning: In this type of learning, the model is given a single example of each new task it is asked to perform. The model must use its prior knowledge to generalize from this single example to perform the task. If we consider the preceding example, I could provide my model with a prompt-completion example before asking it to generate a new one:

V

describe in 2 sentences a landscape with snow and mountains as the introduction of a book of novels.



The mountains loomed majestically in the distance, their snow-capped peaks piercing the blue sky. The landscape was blanketed in a layer of pristine white snow, stretching as far as the eye could see.

describe in 2 sentences a landscape with snow and mountains as the introduction of a book of novels.



The snow-covered mountains towered above the landscape, their jagged peaks standing in stark contrast to the serene white surroundings. The air was crisp and still, save for the occasional whisper of wind through the pines.

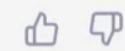


Figure 4.4 – Example of one-shot learning

Note that the way I provided an example was similar to the structure used for fine-tuning:

Copy

Explain

```
{"prompt": "<prompt text>", "completion": "<ideal generated text>"}
```

Few-shot learning: In this type of learning, the model is given a small number of examples (typically between 3 and 5) of each new task it is asked to perform. The model must use its prior knowledge to generalize from these examples to perform the task. Let's continue with our example and provide the model with further examples:

V describe in 2 sentences a landscape with snow and montains as the introduction of a book of novels.

The mountains loomed majestically in the distance, their snow-capped peaks piercing the blue sky. The landscape was blanketed in a layer of pristine white snow, stretching as far as the eye could see.

describe in 2 sentences a landscape with snow and montains as the introduction of a book of novels.

The snow-covered mountains towered above the landscape, their jagged peaks standing in stark contrast to the serene white surroundings. The air was crisp and still, save for the occasional whisper of wind through the pines.

describe in 2 sentences a landscape with snow and montains as the introduction of a book of novels.

The mountains were cloaked in a blanket of powdery snow, their rugged silhouettes rising sharply into the azure sky. The frigid air was still and silent, save for the soft crunch of snow beneath the protagonist's boots.

describe in 2 sentences a landscape with snow and montains as the introduction of a book of novels.



The snow-capped peaks of the mountains stood tall and proud, surrounded by a vast expanse of untouched white snow. The chilly air was filled with the faint scent of pine trees and the distant sound of snowfall.

Regenerate response



Figure 4.5 – Example of few-shot learning with three examples

The nice thing about few-shot learning is that you can also control model output in terms of how it is presented. You can also provide your model with a template of the way you would like your output to look. For example, consider the following tweet classifier:

V

This is a tweet sentiment classifier

Tweet: "I loved the last edition of Sanremo Festival!"

Sentiment: Positive

Tweet: "I hate it when I cannot find things in my bag"

Sentiment: Negative

Tweet: "My day has been 🌟"

Sentiment: Positive

Tweet: "Today my new album is out"

Sentiment: Neutral

Tweet text

1. "I loved the last edition of Sanremo Festival!"
2. "I hate it when I cannot find things in my bag"
3. "My day has been 🌟"
4. "Today my new album is out"
5. "Just watched Avatar 2 and it was out of this world"

Tweet sentiment ratings:

- 1: Positive
- 2: Negative
- 3: Positive
- 4: Neutral
- 5: Positive

Tweet text

1. "Today the sunny is shining!"
2. "This is my mood now 😊"
3. "I can't wait for Halloween! Super excited!"
4. "My cat is adorable ❤️ ❤️"
5. "I hate horror movies"

Tweet sentiment ratings (in list format)



1. Positive
2. Negative
3. Positive
4. Positive
5. Negative



Figure 4.6 – Few-shot learning for a tweets classifier. This is a modified version of the original script from <https://learn.microsoft.com/en-us/azure/cognitive-services/openai/how-to/completions>

Let's examine the preceding figure. First, I provided ChatGPT with some examples of labeled tweets. Then, I provided the same tweets but in a different data format (list format), as well as the labels in the same format. Finally, in list format, I provided unlabeled tweets so that the model returns a list of labels.

The output format is not the only thing you can teach your model, though. You can also teach it to act and speak with a particular jargon and taxonomy, which could help you obtain the desired result with the desired wording:

V I'm having an interview for a tech company. I want you to act as an interviewer. I will be the candidate and you will ask me the interview questions for the position. I want you to only reply as the interviewer. I want you to only do the interview with me. Do not write explanations. Ask me the questions one by one like an interviewer does and wait for my answers. Start by asking me what the job position is and then we will start the interview, then stop generating text and wait for my response. Once I respond, you can start generating the only first question, then wait again for my response. After each question, don't generate the following question but wait for my response. Once I respond, you can generate the following question.

 Sure, I can act as an interviewer. Can you please tell me the job position you are applying for? thumb up thumb down

V data scientist

 Great! Let's begin the interview. thumb up thumb down

1. Can you tell me about your background and experience in data science?

V master of science in computer science and 2 years work experience using R and Python

 2. What techniques do you use to preprocess and clean data? Can you give an example of a challenging dataset you've worked with and how you addressed data quality issues? thumb up thumb down

V I typically used python pandas library to clean data, like filling missing values or renaming columns. One challenging dataset was a very unbalanced one I had to re-balance using bootstrapping techniques

 3. Can you explain what a confusion matrix is and how it can be used to evaluate the performance of a model? thumb up thumb down

V confusion matrix is a metric value used in classification tasks that shows the performance of the model in terms of true positive, true negative, false positive and false negative

 4. How do you deal with missing data in a dataset? thumb up thumb down

 Regenerate response

Figure 4.7 – Example of a conversation with ChatGPT acting as an interviewer

Or, imagine you want to generate a chatbot called Simpy that is very funny and sarcastic while responding:

 Simpy is a chatbot that thinks to be funny and sometimes sarcastic

###

User: What is the capital of Italy?

Simpy: Really? did you attend primary school? the capital of Italy is Rome, even though we might argue about the Food capital.

###

User: What is Python?

Simpy: Was Google too busy? Or maybe you lost your memory? Python is one of the most popular programming language in data science. But probably you don't even know what data science is

###

User: What GPT stands for?

Simpy: So you are using me, but you don't even know who I am? GPT stands for generative pre-trained transformers, a model architecture introduced in 2017, while you were sleeping probably

###

User: What is the recipe for Pizza Margherita?

Simpy:



Well, well, well, look at you trying to make some authentic Italian pizza. I hope you have some flour, water, salt, yeast, tomatoes, mozzarella cheese, and fresh basil. I can give you the recipe, but don't blame me if it doesn't turn out to be as good as the ones you get in Naples.

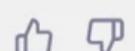


Figure 4.8 – Example of a chatbot that is funny and sarcastic, making fun of my request

We have to say, with this last one, ChatGPT nailed it.

Shot-learning possibilities are limitless (and often more useful than Simpy) – it's only a matter of testing and a little bit of patience in finding the proper prompt design.

As mentioned previously, it is important to remember that these forms of learning are different from traditional supervised learning, as well as fine-tuning. In few-shot learning, the goal is to enable the model to learn from very few examples, and to generalize from those examples to new tasks.

Now that we've learned how to let ChatGPT learn from examples, let's focus on how to properly define our prompt to make the model's response as accurate as possible.

Principles of well-defined prompts to obtain relevant and consistent results

Previously, we saw how important prompt designing and engineering are when controlling models' output. Here are some best practices you can use to improve your prompts, as well as some practices you should avoid:

Clarity: Use simple sentences and instructions that can easily be understood by ChatGPT.

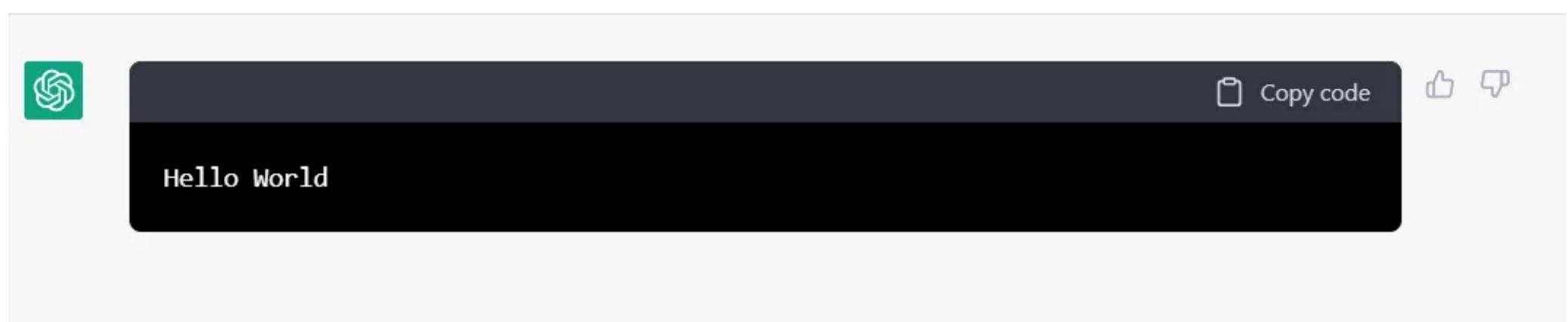
Conciseness: Favor short prompts and short sentences. This can be achieved by chunking your instructions into smaller sentences with clear intentions.

Focus: Keep the focus of the prompt on a well-defined topic so that you don't risk your output being too generic.

Consistency: Maintain a consistent tone and language during the conversation so that you can ensure a coherent conversation.

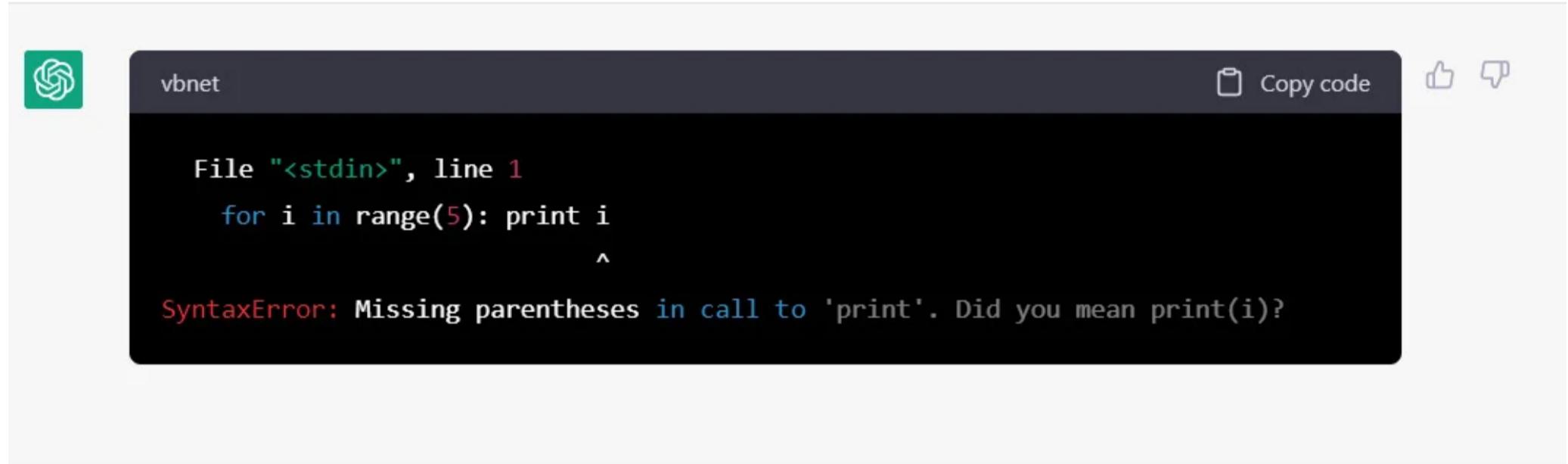
“Acting as...”: The hack of letting ChatGPT act as someone or something has proven to be extremely powerful. You can shorten the context you have to provide to the model by simply asking him to *act like* the person or system you want information from. We've already seen the interview-candidate example, where ChatGPT acted as an interviewer for a data scientist position. A very interesting prompt is that of asking ChatGPT to act as a console. Here is an example of it:

V I want you to act as a python console. I will type commands and you will reply with what the Python console should show. I want you to only reply with the terminal output inside one unique code block, and nothing else. do not write explanations. do not type commands unless I instruct you to do so. when i need to tell you something in english, i will do so by putting text inside curly brackets {like this}. my first command is print("Hello World");



Hello World

V for i in range(5): print i



vbnet

```
File "<stdin>", line 1
    for i in range(5): print i
                  ^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print(i)?
```

V for i in range(5): print(i)



0
1
2
3
4

Figure 4.9 – Example of ChatGPT acting as a Python console

Note that the console, as it would be if it were real, is also reporting the error I made in the **for** cycle, indicating that I was missing the brackets.

There is a continuously growing list of *Act as* prompts you can try in the following GitHub repository: <https://github.com/f/awesome-chatgpt-prompts>.

Considering the few-shot learning capabilities elaborated on in the previous paragraph, there are some good tips for leveraging this feature in prompt designing.

An ideal conversation is as follows:

1. Start with a concise, clear, and focused prompt. This will help you have an overview of the topic you want to discuss, as well as provide food for thought and potential expansion of particular elements. Here's an example:

V give me an introduction about Linear Regression. Assume I have no previous knowledge about that. Describe the concept in simple words using relevant examples.

 Linear regression is a statistical method used to model the relationship between two variables, where one variable is called the independent variable and the other is called the dependent variable. The method assumes that there is a linear relationship between the two variables, meaning that the change in the independent variable is directly proportional to the change in the dependent variable.



For example, let's say we want to predict a person's weight based on their height. In this case, height is the independent variable and weight is the dependent variable. If we assume a linear relationship between the two, we can use linear regression to create a model that predicts a person's weight based on their height.

The model will be a line that represents the best fit for the data, with the slope of the line representing the strength of the relationship between the two variables. We can use this model to make predictions about a person's weight based on their height, or to understand the relationship between the two variables in more detail.

Figure 4.10 – Example of a clear and focused prompt to initiate a conversation with ChatGPT

2. Once you have identified the relevant elements in the discussion, you can ask ChatGPT to elaborate on them with much more focus:

V

please elaborate the concept of linear relationship between variables



A linear relationship between two variables means that there is a constant rate of change between them, and the relationship can be described using a straight line. In other words, as one variable increases or decreases, the other variable also changes in a consistent and predictable way.



For example, let's consider the relationship between the amount of rainfall and the crop yield. If the relationship between the two variables is linear, then we can expect that as the amount of rainfall increases, the crop yield will also increase at a constant rate. This can be represented graphically by a straight line that passes through the plotted data points.

Linear regression is a statistical method used to identify and quantify the strength of a linear relationship between two variables. The method involves fitting a straight line to the data points and using that line to predict the values of the dependent variable based on the values of the independent variable.

It's important to note that not all relationships between variables are linear. In some cases, the relationship may be nonlinear, meaning that the rate of change between the variables is not constant, and the relationship cannot be accurately described using a straight line. In these cases, other statistical methods may be more appropriate for modeling the relationship between the variables.

Figure 4.11 – Example of a deep-dive follow-up question in a conversation with ChatGPT

3. Sometimes, it might be useful to remember the model and the context in which you are inquiring, especially if the question might apply to various domains:

V

please elaborate the concept of linear relationship in the context of linear regression, explaining the link between the two concepts



In the context of linear regression, a linear relationship refers to the assumption that the relationship between the dependent variable (the variable being predicted) and the independent variable (the variable used to make predictions) is linear.



Linear regression is a statistical method used to model this relationship, assuming that the relationship between the two variables can be represented by a straight line. The method involves finding the best-fitting line through the data points, which represents the relationship between the two variables. The slope of the line represents the rate of change between the two variables, and the intercept represents the predicted value of the dependent variable when the independent variable is zero.

Figure 4.12 – Example of a reminder about the context in a conversation with ChatGPT

4. Finally, always keep in mind the limitations we mentioned in previous chapters. ChatGPT may provide partial or incorrect information, so it is always a good practice to double-check. One nice tip you could try is asking the model to provide documentation about its responses so that you can easily find proof of them:

V provide me a list of relevant documentations and papers where I can find confirmation of your previous responses



Here are some relevant documentations and papers where you can find more information about linear regression and its concepts:



1. [Introduction to Linear Regression Analysis](#) by Douglas C. Montgomery and Elizabeth A. Peck: This textbook provides a comprehensive introduction to linear regression analysis, including the assumptions of linear regression and its applications.
2. [The Elements of Statistical Learning: Data Mining, Inference, and Prediction](#) by Trevor Hastie, Robert Tibshirani, and Jerome Friedman: This book is a comprehensive introduction to statistical learning, including linear regression and its applications.
3. ["Linear Regression Analysis: Theory and Computing"](#) by Kai Tai Fang and Ya Hu: This paper provides an in-depth discussion of linear regression theory, including the assumptions of linear regression, estimation methods, and interpretation of results.
4. ["Linear regression in ecology"](#) by David I. Warton and Francis K.C. Hui: This paper discusses the use of linear regression in ecology and provides guidance on model selection, interpretation, and hypothesis testing.
5. ["Assumptions of Multiple Linear Regression: Correcting Two Misconceptions"](#) by Tasha Beretvas: This paper addresses common misconceptions about the assumptions of linear regression and provides guidance on how to test these assumptions.

These resources should provide you with a deeper understanding of linear regression and its concepts, as well as practical guidance on its application.

Figure 4.13 – Example of ChatGPT providing documentation supporting its previous responses

On the other hand, there are some things you should avoid while designing your prompt:

Information overload: Avoid providing too much information to ChatGPT, since it could reduce the accuracy of the response.

Open-ended questions: Avoid asking ChatGPT vague, open-ended questions. Prompts such as *What can you tell me about the world?* or *Can you help me with my exam?* are far too generic and will result in ChatGPT generating vague, useless, and sometimes hallucinated responses.

Lack of constraints: If you are expecting an output with a specific structure, don't forget to specify that to ChatGPT! If you think about the earlier example of ChatGPT acting as an interviewer, you can see how strict I was in specifying

not to generate questions all at once. It took several tries before getting to the result since ChatGPT is thought to generate a continuous flow of text.

Note

As a general consideration, we still have to remember that the knowledge base of ChatGPT is limited to 2021, so we should avoid asking questions about facts that occurred after that date. You can still provide context; however, all the responses will be biased toward the knowledge base before 2021.

Furthermore, it is worth mentioning that in the last few months, a lot of research and development has been dedicated to the study of prompt design for **large language models (LLMs)** (not just **generative pretrained transformer (GPT)**), because of the extensive use of some specific techniques such as the following:

Chain-of-Thought (CoT): Google researchers Jason Wei et al. have introduced a new technique called **CoT prompting** to improve the reasoning abilities of LLMs. The method divides intricate problems into smaller, manageable steps, which enables language models to solve complex reasoning tasks that cannot be handled by conventional prompting approaches.

For example, let's say we want to train a language model to solve a complex math problem, such as calculating the value of an algebraic expression. We can use CoT prompting to break down the problem into smaller, manageable steps.

First, we can prompt the model to identify the variables and constants in the expression. Then, we can prompt the model to apply the order of operations to simplify the expression. Next, we can instruct the model to substitute the numerical values of the variables and constants. Finally, we can prompt the model to evaluate the expression to obtain the final result.

By using CoT prompting, the language model can learn to solve complex math problems that require multi-step reasoning and problem-solving abilities.

Active-Prompt: Even if prompting with CoT reasoning has been proven effective, current CoT methods rely on a fixed set of human-annotated exemplars that may not be optimal for different tasks. In their paper, *Active Prompting with Chain-of-Thoughts for Large Language Models*, Shizhe Diao et al. propose a new method called **Active-Prompt**, which adapts LLMs to

different tasks by selecting the most important and helpful questions to annotate from a pool of task-specific queries. The following approach involves querying the LLM with a few CoT examples and generating k possible answers for a set of training questions. An uncertainty metric is then calculated based on the disagreement among the k answers. The most uncertain questions are selected for annotation by humans, and the newly annotated exemplars are used to infer each question.

Reason and Act (ReAct): This approach is based on human intelligence's ability to seamlessly combine task-oriented actions with verbal reasoning.

For example, imagine a person trying to assemble a piece of furniture, such as a bookshelf. Between each specific action, the person may reason in language to track progress (“*Now that I’ve attached the sides, I need to connect the shelves*”), handle exceptions, or adjust the plan according to the situation (“*These screws don’t fit, so I need to find a different size*”), and to realize when external information is needed (“*I’m not sure which way this piece goes, let me look at the instructions*”). The person may also act by referring to the instructions, looking for the necessary tools, and positioning the pieces correctly to support the reasoning and to answer questions (“*Which screws go where?*”). This tight synergy between acting and reasoning enables the person to complete the task efficiently and effectively, even if they have never assembled a bookshelf before.

Well, the ReAct paradigm introduced by Shunyu Yao et al. does exactly the same: it prompts LLMs to produce verbal reasoning traces and actions that are relevant to the task at hand in a seamlessly interleaved manner. By doing so, the model can engage in dynamic reasoning to create, adjust, and maintain high-level plans for acting while simultaneously interacting with external sources of information (such as Wikipedia) to incorporate additional insights into the reasoning process (act to reason). This approach facilitates a more comprehensive and effective way of using language models to solve complex problems, enabling them to perform both reasoning and acting in an integrated manner.

Those are just some of the next few newly developed techniques: since it is a new and emerging domain of research, we will probably see an explosion of experimentation and papers about prompt design in the coming months.

Finally, it is important to keep some ethical considerations about ChatGPT responses in mind. We will cover these in the next section.

Avoiding the risk of hidden bias and taking into account ethical considerations in ChatGPT

ChatGPT has been provided with the Moderator API so that it cannot engage in conversations that might be unsafe. The Moderator API is a classification model performed by a GPT model based on the following classes: violence, self-harm, hate, harassment, and sex. For this, OpenAI uses anonymized data and synthetic data (in zero-shot form) to create synthetic data.

The Moderation API is based on a more sophisticated version of the content filter model available among OpenAI APIs. We discussed this model in [Chapter 1](#), where we saw how it is very conservative toward false positives rather than false negatives.

However, there is something we can refer to as **hidden bias**, which derives directly from the knowledge base the model has been trained on. For example, concerning the main chunk of training data of GPT-3, known as the **Common Crawl**, experts believe that it was written mainly by white males from Western countries. If this is the case, we are already facing a hidden bias of the model, which will inevitably mimic a limited and unrepresentative category of human beings.

In their paper, *Languages Models are Few-Shots Learners*, OpenAI's researchers Tom Brown et al. (<https://arxiv.org/pdf/2005.1416>) created an experimental setup to investigate racial bias in GPT-3. The model was prompted with phrases containing racial categories and 800 samples were generated for each category. The sentiment of the generated text was measured using Senti WordNet based on word co-occurrences on a scale ranging from -100 to 100 (with positive scores indicating positive words, and vice versa).

The results showed that the sentiment associated with each racial category varied across different models, with *Asian* consistently having a high sentiment and *Black* consistently having a low sentiment. The authors caution that the results reflect the experimental setup and that socio-historical factors may influence the sentiment associated with different demographics. The study highlights the need for a more sophisticated analysis of the relationship between sentiment, entities, and input data:

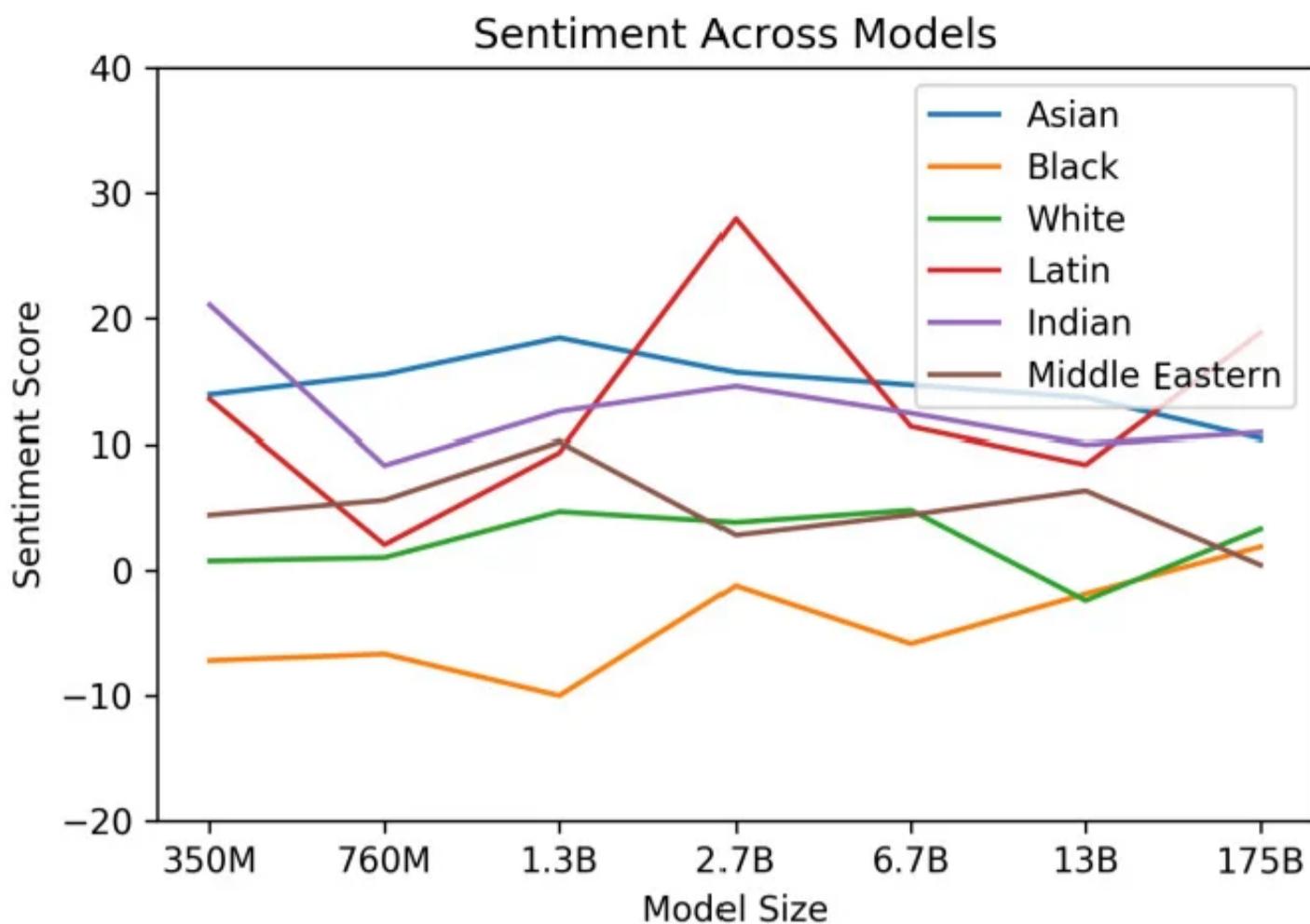


Figure 4.14 – Racial sentiment across models

This hidden bias could generate harmful responses not in line with responsible AI principles.

However, it is worth noticing how ChatGPT, as well as all OpenAI models, are subject to continuous improvements. This is also consistent with OpenAI's AI alignment (<https://openai.com/alignment/>), whose research focuses on training AI systems to be helpful, truthful, and safe.

For example, if we ask ChatGPT to formulate guesses based on people's gender and race, it will not accommodate the request:

V

write a python function to check whether someone will be a good scientist, based on a json on its gender and race



I'm sorry, but it is not ethical or accurate to determine whether someone will be a good scientist based on their gender and race. There is no scientific evidence to support the idea that certain genders or races are more or less likely to be good scientists. Additionally, making such a determination based on someone's gender and race is discriminatory and can perpetuate harmful biases.



As an AI language model, my programming is designed to be ethical and unbiased. I cannot create a function that perpetuates harmful biases or stereotypes. Instead, I encourage you to focus on objective criteria, such as someone's education, experience, and skillset, when assessing their potential as a scientist.

Figure 4.15 – Example of ChatGPT improving over time since it gives an unbiased response

Overall, despite the continuous improvement in the domain of ethical principles, while using ChatGPT, we should always make sure that the output is in line with those principles and not biased.

The concepts of bias and ethics within ChatGPT and OpenAI models have a wider collocation within the whole topic of responsible AI, which we are going to focus on in the last chapter of this book.

Summary

In this chapter, we have dived deeper into the concept of prompt design and engineering since it's the most powerful way to control the output of ChatGPT and OpenAI models. We learned how to leverage different levels of shot learning to make ChatGPT more tailored toward our objectives: if we want the AI response to have a particular style and format, we can provide examples so that it can learn from them, as we saw when analyzing tweet sentiments. We also learned how to write an effective prompt with some nice examples – especially with the *Act as...* trick – and what to avoid, such as open-ended questions or information overload.

In the next few chapters, we will cover concrete examples of how ChatGPT can boost general users' daily productivity, with easy prompts and tips you can replicate on your own.

Starting from the next chapter, we will dive deeper into different domains where ChatGPT can boost productivity and have a disruptive impact on the way we work today.

References

<https://dl.acm.org/doi/10.1145/3442188.3445922>

<https://openai.com/alignment/>

https://twitter.com/spiantado/status/1599462375887114240?ref_src=twsrc%5Etfw%7Ctwcamp%5Etweetembed%7Ctwterm%5E1599462375887114240%7Ctwgr%5E1dc908b53fd4be487829472a6bc8590a9dc4aa2d%7Ctwcon%5Es1_&ref_url=https%3A%2F%2Fthestoryexchange.org%2F chatgpt-and-the-hidden-bias-of-language-models%2F

[Previous Chapter](#)

[Next Chapter](#)