ChatGPT as a Cobot: ChatGPT versus davinci-instruct

by Denis Rothman and ChatGPT

In this article, I used ChatGPT as a cobot (a collaborative bot). Essentially, ChatGPT was my co-writer for this article.

We (ChatGPT and I) will go through OpenAI davinci-instruct beta, ChatGPT versus davinci-instruct, and explore the lengths of conversational responses.

Before we explore how ChatGPT can be used as a cobot, we'd love your thoughts on how we can improve your learning experience. As a token of appreciation, we're offering a free preview of our exciting new product to 25 participants of the survey.

Register me for the FREE preview

Let's get started.

Denis Rothman: How can ChatGPT be a cobot for programmers?

ChatGPT: As an AI language model, ChatGPT can be a useful tool for programmers in a number of ways. Here are some examples:

Code assistance: ChatGPT can be trained on a vast amount of programming language and syntax, which makes it an ideal companion for programmers who need assistance with coding tasks. It can provide suggestions on code structure, syntax, and even point out errors in code, which can save programmers time and improve the quality of their code.

Learning: ChatGPT can also be trained on educational materials related to programming, making it a valuable resource for programmers who want to learn new languages, frameworks, or programming concepts. Programmers can ask ChatGPT questions, receive explanations, and even practice coding in a safe and controlled environment.

Collaboration: ChatGPT can be used as a collaborative tool for programmers, helping them to communicate and work together more effectively. Programmers can ask ChatGPT to facilitate discussions, share code snippets, and even translate between languages to bridge the gap between different teams or cultures.

Overall, ChatGPT can be a valuable Cobot for programmers, helping them to work more efficiently, learn more effectively, and collaborate more seamlessly.

> **Note**: *Transformers for Natural Language Processing, 2nd Edition*, *Chapter 7*, *The Rise of Suprahuman Transformers with GPT-3 Engines* describes how to use GPT-3.

Let's import `openai` and use our API key to use davinci-instruct:

```
#Importing openai
try:
  import openai
except:
  !pip install openai
  import openai
```

```
#Store you key in a file and read it(you can type it directly in the notebook but
it will be visible for somebody next to you)
from google.colab import drive
drive.mount('/content/drive')
f = open("drive/MyDrive/files/api_key.txt", "r")
API_KEY=f.readline()
f.close()

#OpenAI Key
import os
os.environ['OPENAI_API_KEY'] =API_KEY
openai.api_key = os.getenv("OPENAI_API_KEY")

Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

Now let’s begin working with davinci-instruct.

1. OpenAI davinci-instruct-beta

At this point, I wanted more information on the `max_tokens` parameter which controls the length of the prompt and the response.

I used **ChatGPT** as a cobot. A dialog with ChatGPT on this topic provided the following insights and tips.

Denis: What is the maximum number of tokens for the davinci-instruct model?

ChatGPT: As of my knowledge cutoff date of September 2021, the maximum number of tokens for the davinci-instruct model of OpenAI's GPT-3 language model was 2048 tokens. This means that the model can take in a sequence of up to 2048 tokens as input, with each token representing a word, symbol, or punctuation mark.

However, it's worth noting that the exact specifications and capabilities of OpenAI's models can change over time, so it's possible that the maximum number of tokens for the davinci-instruct model may have been updated since my knowledge cutoff. It's always best to consult the most up-to-date documentation from OpenAI for the latest information.

Denis: When I set up davinci-instruct for 2000 tokens, it only replies with a very short answer. How can I force davinci-instruct to provide longer responses?

ChatGPT: While the davinci-instruct model can accept input sequences of up to 2048 tokens, the length of the response you receive may depend on a variety of factors, such as the complexity of the input prompt, the specific task or prompt you are using, and the level of detail required in the response.

If you are finding that the model is only providing very short answers to your prompts, there are a few strategies you can try to encourage it to provide longer responses:

Provide more detailed input: The davinci-instruct model works by predicting the most likely next word in a given sequence based on the context and content of the input it has received. If you provide more detailed input that gives the model a clearer idea of what you are looking for, it may be more likely to generate a longer, more informative response.

Increase the temperature: The temperature parameter controls the level of randomness and creativity in the model's responses. If you increase the temperature, the model may be more likely to generate longer and more diverse responses. However, be aware that increasing the temperature can also lead to more nonsensical or irrelevant responses.

Use a different prompt or task: If you are consistently getting very short answers from the model, it may be helpful to experiment with different prompts or tasks to see if you can get a more robust response. For example, you might try asking the model to generate a longer piece of text, or to answer a series of related questions rather than a single prompt.

It's worth noting that the davinci-instruct model is a highly complex and sophisticated language model, and there is no guaranteed way to force it to generate longer or more detailed responses. However, by experimenting with different input strategies and parameters, you may be able to find a configuration that works better for your specific needs.

Denis: Thank you.

ChatGPT: You're welcome! If you have any more questions, feel free to ask.

Now, let’s set the parameters for the model, including the prompt:

```
#GPT-3 parameters, p2=the prompt as for ChatGPT
p1="davinci-instruct-beta"
p2="Summarize the Critique of Pure Reason:"
p3=0
p4=2000
p5=1
p6=0
p7=0

response =
openai.Completion.create(engine=p1,prompt=p2,temperature=p3,max_tokens=p4,top_p=p
5,frequency_penalty=p6,presence_penalty=p7)
r = (response["choices"][0])
print(r["text"])


The Critique of Pure Reason is a book by Immanuel Kant that is a critique of the
limits of human understanding. Kant argues that the human mind is limited in its
ability to know the world as it really is.
```

You can continue the conversation by adding a new prompt. This time we’ve reduced the number of tokens from 2,000 to 120:

```
#We continue the dialog
p1="davinci-instruct-beta"
p2=" In the Critique of Pure Reason Kant argues that the human mind is limited in
its ability to know the world as it really is. Explain what the limits of the
human mind are:"
p3=0
p4=120
p5=1
p6=0
p7=0

response =
openai.Completion.create(engine=p1,prompt=p2,temperature=p3,max_tokens=p4,top_p=p
5,frequency_penalty=p6,presence_penalty=p7)
r = (response["choices"][0])
```

```
print(r["text"])


The human mind is limited in its ability to know the world as it really is.
```

Impressive results! Now let’s compare this with what ChatGPT can do.

2. ChatGPT versus davinci-instruct-beta

The following is the output of ChatGPT for the prompt in this article.

Denis: Summarize the Critique of Pure Reason by Kant

ChatGPT(excerpt of the response): Immanuel Kant's Critique of Pure Reason is a complex and influential philosophical work that aimed to establish the limits of human knowledge and understanding.

Kant's central argument in the Critique of Pure Reason is that our understanding of the world is not solely derived from experience but is also shaped by the inherent structure and organization of our minds. He claimed that our minds impose categories and concepts onto the raw data of experience, which allows us to make sense of it but also limits our knowledge to what we can conceptualize.

.../...

Overall, the Critique of Pure Reason is a complex and influential work that established many of the key ideas of modern philosophy, including the limits of human knowledge, the importance of empirical investigation, and the role of reason in ethical principles.

ChatGPT provides a more detailed answer. But why?

3. ChatGPT cobot on the length of responses

Davinci-instruct-beta provided short answers, while ChatGPT provided a longer response. In some cases, a conversational chatbot should provide *short* answers and in other cases, *long* answers would be better (depending on the situation). If you're looking for the capital of a country, for example, you'll only need a short answer. If you want an explanation on a topic, you might want a longer answer. This can be controlled with the `max_tokens` parameter.

I asked my ChatGPT cobot about this:

Denis: Why are ChatGPT's answers longer than davinci-instruct responses for the same question?

ChatGPT: There could be several reasons why ChatGPT's answers are longer than responses generated by the davinci-instruct model for the same question.

Firstly, it's important to note that ChatGPT and the davinci-instruct model are different models with different architectures, training data, and parameters. While both models are based on the GPT family of language models, they have been trained on different datasets and have different strengths and weaknesses.

In general, ChatGPT may be more focused on generating longer and more detailed responses because it has been specifically designed to generate conversational responses to a wide variety of prompts. As a language model trained for conversation, it may be more likely to generate longer and more complex responses in order to simulate a natural conversation and provide a more complete answer.

On the other hand, the davinci-instruct model has been designed specifically for generating instructional content and providing step-by-step guidance for various tasks. As such, it may be more focused on generating responses that are concise and to the point, while still providing all the necessary information to complete the task at hand.

Additionally, the specific input prompt or question that you provide to each model can also have an impact on the length of the response you receive. If you provide a more detailed or open-ended prompt to ChatGPT, it may generate a longer response in order to provide a more complete answer. On the other hand, if you provide a more focused or task-specific prompt to the davinci-instruct model, it may generate a shorter response that is more directly relevant to the task at hand.

Overall, the length and complexity of the responses generated by different language models can depend on a wide variety of factors, including the specific model architecture, training data, parameters, and input prompts.

Denis: Thanks!

ChatGPT: You're welcome! Let me know if you have any more questions.

There we have it, how to use ChatGPT as a cobot. If you'd like to learn more about ChatGPT and GPT-4, then check out the GitHub repository for the `Transformers for Natural Language Processing, 2nd Edition` book: `https://github.com/Denis2054/Transformers-for-NLP-2nd-Edition`.

Don't forget to fill in our survey, as we'd love your thoughts on our new product that we're developing. As a token of our appreciation, we're offering a free preview of our exciting new product to 25 participants of the survey.

Register me for the FREE preview!