Learning Objectives

Learners will be able to...

- **Generate answers using OpenAI**
- **Practice with classification**
- **Learn about Generation**
- **Compare translation to transformation**
- **Run different code generated by GPT-3**
- **Discuss the deployment process (e.g. safety review)**

info

Make Sure You Know

- You are familiar with Python.
- You can generate a response using `openai.Completion.create`.

Question Answering

We have seen how GPT-3 answers questions based on existing knowledge. What happens if we ask it a question that it could not know based on existing data? Copy the question below and paste it into the file on the left. Click the `TRY IT` button to see the response, which appears in the file on the left.

```
Q: What is my name?
```

You can use the reset button below to clear the text in the file on the left.

Notice how it tells you some variation of it does not know. We are going to show the model a pattern to use instead of saying "I don't know". Guide the model towards a factual answering by showing it how to respond to questions that fall outside its knowledge base. Using a `?` to indicate a response to words and phrases that it does not know provides a natural response that seems to work better than more abstract replies. Copy the entirety of the block below and paste into your box.

```
Q: Who is james bond?
A: James Bond is a fictional character created by Ian Fleming.
        Bond is a British secret agent who works for MI6.

Q: what is a lkdaos?
A: ?

Q: who was Barack Obama?
A: Barack Obama was the 44th President of the United States.

Q:what is ploasd?
```

After running the code above try running a question it would know, such as the one below. GPT-3 responds with the correct answer because you are only training the model to respond with a `?` when it does not know the answer.

```
Q:what planet is the biggest in our solar system?
```

GPT-3 also has the ability to generate answers for questions that require a complex response. For example, you can ask the model to return ten movies from the science fiction genre.

```
List 10 science fiction movies:
```

You can use the reset button to clear the text file and create new and complex questions for GPT-3. Use the `TRY IT` button to generate another response.

Classification

GPT-3 can classify items into categories. Preface your prompt with instructions to classify a list of words.

```
classify the following : cat, dog , car , plane
```

The model can use pattern matching to format the output of the classifications. The prompt below first asks GPT-3 to classify the list of objects. The next two lines are the pattern. List the object from the prompt and, on a new line, print `category:` followed by the classification of the object.

```
classify the following : cat, dog , car , plane
cat
category: animal
```

The pattern matching above is similar to the way we guided the model in responding to questions that fall outside its knowledge base.

You can also ask GPT-3 to replace its own classification system with another. The examples below prompt the model to use the Entertainment Software Ratings Board (ESRB) rating system with an accompanying text. Compare the ratings between the two examples.

▼ **What is the ESRB?**
The ESRB is an organization that rates video game content (in Canada and the US) according to age and content. They use a system similar to the motion picture rating system.

```
Provide an ESRB rating for the following text:

"There was once a great ninja who lived in a small village in
        Japan. He was a master of all the ninja arts and was
        respected by all who knew him. One day, a rival ninja
        from a nearby village challenged him to a duel. The
        ninja accepted and they fought a fierce battle."
```

```
Provide an ESRB rating for the following text:

"Once upon a time, there was a vampire who lived in a dark, dank
        castle. He was a handsome vampire, with a strong jaw and
        piercing blue eyes. But he was also a cold-blooded
        killer, and he enjoyed nothing more than sinking his
        teeth into their neck and drinking the blood of his
        victims."
```

The GPT-3 model is adaptive in that it can use a third-party rating system for classification purposes.

Generation

Up until this point, we have used the GPT-3 model as a means to answer questions or classify words. One of the more interesting features associated with OpenAI is its ability to generate new and innovative content. For example, you can create a prompt that asks the model to generate three different product descriptions.

```
Create 3 product names for the following description
a laptop that can last you 20 years
```

challenge

Try this variation:

Have the model create two taglines for a donut shop.

```
write 2 taglines for a donut shop
```

The generation is not limited to simply text. You can ask OpenAI to generate emoji based on text in the prompt. In this example, we are asking the model to turn movie titles into emoji.

```
Convert these movie titles into emoji: Matrix, Mulan, Spy kids
```

Not only can GPT-3 produce emoji, but it can also do the inverse. You can ask the model to generate movie titles based on a sequence of emoji.

```
Based on the following guess the movie title :     ,     , 🦈 ♀
```

Lastly, the GPT-3 model can work with prompts that are more open-ended in nature. There should be multiple responses told from two different points of view, and they should still remain coherent.

```
Emulate a text message conversation.
```

challenge

Try these variations:

Give the conversation a topic (some good news) and have the conversation last longer.

```
Emulate a long text message conversation about some good
        news.
```

Translation and Transformation

In addition to text generation, the GPT-3 model can translate and transform text.

Translation

Translation is the process of converting information from one form to another. In the example below, we are asking the model to convert a phrase from English into French.

```
translate the following to French: I am hungry and I want some
        pizza.
```

In addition, the model can figure out a translation without having to be told the specific language. In the example below, it can determine that a phrase in Spanish needs to be converted into English.

```
what does "Me gustaría comer una pizza" mean
```

Translation is not just going from one language to another. We can ask the model to translate a text from first- to second- or third-person. The following example translates the phrase from first-person to third-person.

```
Convert first-person to the third-person: "I am big eater. I
        like to eat pizza in my car"
```

challenge

Try this variation:

Translate the phrase from present tense to past tense.

```
convert the following to past tense: I go to the store
```

Transformation

Transformation has a slight but important difference from translation. Transformation is the process of converting information from one structure or format into another. In the example below, we give the model a sentence that is grammatically incorrect. GPT-3 can transform it into correct English.

```
Correct sentences into standard English
I'm go to hunt for food. I no went to the park.
```

We can transform a text into a version that is easier to understand. The following example asks the model to simplify the definition of a function so a second grader can understand it.

```
simplify the following text for a 2nd grader:

Functions are a sequence of instructions packaged as unit that
        perform a specific task. Programming languages come with
        pre-defined functions in their standard library. You can
        also create your own user-defined functions.
```

We can use OpenAI to transform from one programing language to another. The following example converts a simple for loop in Python to its equivalent in JavaScript.

```
convert python to javascript

for i in range (0,8):
    i=i+1
    print(i+2)
```

Code Generation and Translation

We tend to think about GPT-3 text generation abilities as being related to human languages. However, it can also work with programming languages. In the example below, the model can reason about a given code sample and determine the end result.

```
what is the result of the following code?
x = 3
print(x ** 2)
```

```
generate python code to sort an unsorted list
```

Because GPT-3 can reason about code, then it should be able to find bugs in the code.

```
# Fix bugs in the python function

# Buggy Python
import Random
a = "12"
b = random.randint(1,12)
add(a,b)
```

We can use it to explain a piece of Python code in human language.

```
explain the following code
import Random
a = "12"
b = random.randint(1,12)
add(a,b)
```

More examples can be found in the OpenAI website.

Deployment Process

One of the breakthroughs with the GPT-3 model is the generation of text that seemingly comes from a human. While there are many obvious benefits to this, there are also many obvious risks. OpenAI released a paper in which they dedicate an entire section to the broader impacts of this technology. Specifically, they talk about the misuse of language models; fairness, bias, and representation; and energy usage.

OpenAI recommends several key principles to help providers of large language models (LLMs) mitigate the risks of this technology in order to achieve its full promise to augment human capabilities. As LLM providers, these principles are published in order to represent a first step in collaboratively guiding safer large language model development and deployment.

Prohibit Misuse

- Prohibit material harm to individuals, communities, and society such as through spam, fraud, or astroturfing.
- Build systems and infrastructure to enforce usage guidelines. This may include rate limits, content filtering, application approval prior to production access, monitoring for anomalous activity, and other mitigations.

Mitigate Unintentional Harm

- Proactively mitigate harmful model behavior. Best practices include comprehensive model evaluation to properly assess limitations, minimizing potential sources of bias in training corpora, and techniques to minimize unsafe behavior such as through learning from human feedback.
- Document known weaknesses and vulnerabilities, such as bias or ability to produce insecure code, as in some cases no degree of preventative action can completely eliminate the potential for unintended harm. Documentation should also include model and use-case-specific safety best practices

Thoughtfully Collaborate with Stakeholders

- Build teams with diverse backgrounds and solicit broad input. Diverse perspectives are needed to characterize and address how language models will operate in the diversity of the real world, where if unchecked they may reinforce biases or fail to work for some groups.

- Publicly disclose lessons learned regarding LLM safety and misuse in order to enable widespread adoption and help with cross-industry iteration on best practices.
- Treat all labor in the language model supply chain with respect. For example, providers should have high standards for the working conditions of those reviewing model outputs in-house and hold vendors to well-specified standards (e.g., ensuring labelers are able to opt out of a given task).

Coding Exercise

Training the Model

Train the model so that when it receives a question it cannot answer, it responds with `///`.

When you are ready to check your work, add the nonsensical question below to the prompt. Then use the try it button below to test your answers.

```
Q: what is my np'kmdpfd?
```

The mode should respond in the following manner:

```
Q: what is my np'kmdpfd?
A: ///
```

You can use the reset button below to clear the text in the file on the left.