

API OpenAI

- Utilizamos os mesmos métodos para diferentes tarefas: "few-shot learning."
 - `openai.Completion.create()`
- A tarefa deve ser definida no argumento `prompt`, por exemplo:
 - "Você pode traduzir este texto...."
 - "Você pode gerar perguntas sobre..."
 - "Você pode resumir este texto..."



Modelo

- Devemos definir um modelo pré-treinando que será utilizado
- Existem vários disponíveis para a API
- Modelos possuem características e tamanhos diferentes
- Podem ter desempenhos diferentes para tarefas diferentes



Família de Modelos Baseados em GPT-3

Família

davinci	Mais poderoso, para qualquer tarefa
curie	Rápido
Babbage	Para tarefas específicas, rápido
Ada	Tarefas simples, o mais rápido
Codex	Geração de Código

Alguns modelos: Davinci

- davinci
- text-davinci-002: versão menor e mais rápida
- text-davinci-003: versão melhorada e atualizada do 002, respostas mais elaboradas



Outros modelos

- text-curie-001
- text-babbage-001
- text-ada-001



Codex: Code Davinci

- code-davinci-002
- code-cushman-001: mais rápido



Argumentos: Parte I

- model: modelo
- prompt: texto de entrada.
- temperature: valor entre 0 e 1 que define a aleatoriedade do texto. Quanto mais próximo de 1 mais aleatório. Padrão é 0.7.
- max_tokens: número máximo de palavras na resposta (inclui pontuação)



Argumentos: Parte II

- n: quantidade de respostas geradas
- stop: Lista texto que pode ser utilizado para parar a geração do texto. Padrão é none
- echo: Define se o texto de entrada é incluído na resposta gerada.

