**OpenAI: Usage, Implementation & Fine Tuning for Headline Generation with Key Words**

In this article you will learn about:

1. Overview of OpenAI
2. Features, Applications and Domains of AI
3. OpenAI Playground based Applications
4. Headline Generation and Fine Tuning Using Open AI
5. **Overview of OpenAI**

OpenAI is ensuring the artificial general intelligence (AGI) benefits for the humanity. Starting from 2015 OpenAI has introduced numerous features i.e., Open AI Gym Beta, Universe, Proximal Policy Optimization (PPO), Dota 2, OpenAI LP, MuseNet, GPT-2, JukeBox, DALL-E, GPT-3 and currently introduced DALL-E2.

AI Studio provides consultancy in numerous AI applications and in this article, we will discuss about OpenAI and its application in numerous fields along with its implementation in Python and relevant platform|(s). OpenAI is an AI research and deployment company aiming to ensure that artificial general intelligence benefits all of humanity. The OpenAI API may be used for almost any job involving the interpretation or generation of natural language processing-oriented applications. It has numerous models i.e., GPT-3 with multiple variants i.e., ada that are ideal for various applications. Moreover, it provides the facility to fine-tune existing model as per your own ideas to increase the effectiveness and efficiency of your unique models. It enhances the reliability, robustness, and flexibility of user-oriented application with improved user interaction and experience. These fine-tuned models can be applied on the variety of applications including (not limited to) semantic search, Q&A based models, content creation tools and headline generation tools etc. models may be applied to a variety of tasks, including content creation, semantic search, and categorization.

Diagram

Description automatically generated

Figure 1: Information source OpenAI [1]

1. **Features, Applications and Domain of OpenAI**
   1. ***Features***
2. ***Completion***

This feature can be used to perform variety of tasks related to the generation and manipulation of the text. User inputs some text in the prompt the OpenAI generates text completion related to the user requirements. Upon every attempt, OpenAI playground provides different but related answers. These settings can be adjusted through adjusting playground parameters, i.e., temperature, maximum length, frequency etc. A sample of Open AI playground is given in Figure 2. This feature offers the functionalities of classification, generation, conversations, translations, conversions, summarization, completion, generation of factual responses, text insertion and text editing.

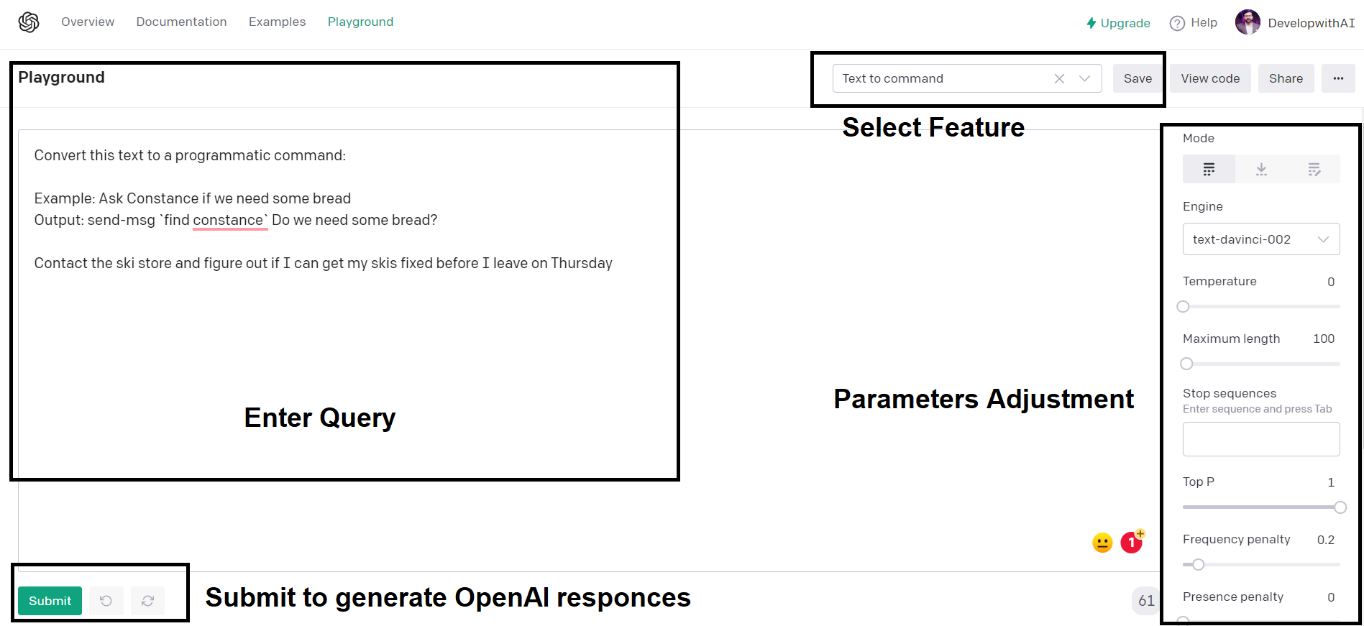


Figure 2: Overview of OpenAI Playground [2]

1. ***Semantic search***

This feature (/search), enable its users to perform semantic search over a wide variety of the articles i.e., 200 documents can be passed using the document parameters of the OpenAI. Basically, semantic search is a data searching technique in which user enters a query to determine the context and intent of the query being inserted by the user. These documents can be words, sentences, paragraphs, or longer relevant documents.

A picture containing text

Description automatically generated

Figure 3: Retrieval Process of OpenAI in Semantic Search [3]

OpenAI has offered beta API where documents can be preloaded and then user can call of those as a file collectively. This case only happens if you have more than 200 documents to be included in the semantic search process.

1. ***Fine-tuning***

Fine-tuning is the most amazing feature of the OpenAI which enables user to customize their models as per the requirements of their applications. It gives the higher quality results then the OpenAI built in prompt design with the ability to train more than one example at a time. Since this provides an opportunity to run multiple examples with the single prompts, resultantly, it reduces the latency requests and saves tokens.

Mainly OpenAI API is powered with three engines GPT-3, Codex and content filter. Here, GPT-3 has the ability to handle natural language. In fine-tunning applications usually, GPT-3 based models are used. Those models are available with a limited number of tokens (several thousands) and are based on training data. Those models can be listed as “text-divinici-002, text-curie-001, text babbage-001, text-ada-001”.

These models can be used by the users as per the application requirements, but these have few cost advantages. Similarly, divinici has the highest performance comparative to other model but other models can also perform better as depending upon the application, where they have been deployed.

PowerPoint

Description automatically generated with low confidence

Figure 4: General Process of Fine tuning in OpenAI [4]

1. ***Classification***

The Classifications endpoint can imported as (/classifications) in prompt; may be used for any text-to-label job and allows users to use a labelled collection of samples without fine-tuning in their applications being developed. It minimizes the requirement for hyper-parameter adjustment by eliminating fine-tuning. The endpoint acts as a "autoML" solution that is simple to set up and react to changes in label schema. At query time, up to 200 labelled samples or a pre-uploaded file can be supplied for further processing.

These queries can be classified as per the requirements of the users.

Diagram

Description automatically generated

Figure 5: General Process of Classification in OpenAI [5]

1. ***Question Answering***

The Q&A prompt is also available in the OpenAI’s beta API. Answers (/answers) is a specialized question-answering endpoint in OpenAI that may be used in applications that demand high-accuracy text production from sources of truth such as business documents and knowledge bases. To go above that limit, the extra context can be given as a list of up to 200 documents or as a pre-uploaded file. If the number of documents exceeds from 200 then it can be uploaded by file.

Diagram

Description automatically generated

Figure 6: General Process of answer endpoint in OpenAI [6]

1. **OpenAI Playground based Applications**

Following is the list of top examples by OpenAI developed for the enhanced user experience.

|  |  |
| --- | --- |
| Q&A | Grammar Correction |
| Summarize to a 2nd Grader | Natural Language to OpenAI API |
| Text to Command | Product Name Generator |
| English to Other Languages | Python Bug Fixer |
| Text to Command | Friend Chat |
| SQL Translate | Mood to Color |
| Factual Answering | TL;DR Summarization |
| ML/AI Language Model Tutor | Science Fiction Book List Maker |

1. Headline Generation and Fine Tuning Using Open AI
2. *Installation of OpenAI in python:*

To install the official Python bindings, run the following command

|  |
| --- |
| pip install openai |

To install the official Node.js library, run the following command in your Node.js project directory:

|  |
| --- |
| npm install openai |

1. Authentication:

To authenticate OpenAI API uses API Keys. These are the personalized keys for the individuals and organizations. To make an API Key follow these steps:

1. Visit <https://beta.openai.com/overview>
2. Click on the top right corner and sign up for OpenAI and follow procedure.
3. After successful sign up, again visit <https://beta.openai.com/overview>
4. Click on the top right corner of the page to log in.
5. Once you have successfully logged in, you will be directed to a new page.
6. Again look at the top right corner of the page and click on you profile picture.
7. A dropdown menu will be shown and click on “View API Keys”
8. Now you can simply copy your personalized API key.
9. Please Note: Do not share your personalized API Key with anyone.
10. *Application: Headline Generator using keywords.*

Outcome: How to implement and utilize the Open AI finetune modeling to have a complete result according to our application.

For Finetuning any model, the first thing to understand is that you use the dataset that you have created yourself to get better results.

This consultancy is divided into four main parts for now, which are discussed below:

1. ***Dataset Formation:***

The data that you currently have is in this format given below:

Current Formation of Data:

[  
{  
"title": "Facebook is dumbest social media platform? ",  
"keywords": [  
"internet",  
"web",  
"social media"  
],  
},  
{  
"title": "We should eat more and sleep less",  
"keywords": [  
"Eat",  
"Sleep",  
"Food"  
]   
}  
{  
"title": "Messi is the greatest player of all times!",  
"keywords": [  
"Football",  
"Leon Messi",  
"Games"  
]   
}  
]

Dataset Formatting:

Now the thing to understand before getting into OPEN AI finetuning is that the dataset is divided into two main parts, one part is the prompt, and the second part is the completion.

According to our application we have keywords, categories and a title is created at the end. Therefore, what we can do here is put the category and keywords in the prompt side and the headline that we have in completion.

Example:

{"prompt":" social media. Internet. Web. ->", "completion":" Facebook is dumbest social media platform? "}

After making csv file covert it to jsonl. The conversion will be discussed int the next step.

1. **Dataset upload and management**

Before uploading the dataset, first you must install OPEN AI and also do the fine tuning by using its API. These steps are discussed here:

1. Open the command window and the environment that you are using and install OPEN AI and wandb. You must also have python 3 installed before this works for you.

pip install openai

pip install --upgrade openai wandb

1. If the data is in the csv file, we must prepare it for open AI finetune file:

openai tools fine\_tunes.prepare\_data -f <LOCAL\_FILE>

example: openai tools fine\_tunes.prepare\_data -f C:\F Disk\data.csv

1. Now in the next step, we can use the developed jsonl file to fine tune our model, for this we have to give a basic model from GPT-3. Its latest model is Davinci so we will use its latest version if possible.

openai api fine\_tunes.create -t <TRAIN\_FILE\_ID\_OR\_PATH> -m <BASE\_MODEL>

example: openai api fine\_tunes.create -t C:\F Disk\data.jsonl -m ada –batch\_size 64

1. **Prompt Formatting and input:**

Now after fine tuning the model, the next step is to use this for our application, for that we have flexibility to use CMD or Python or even node.js:

1. **CMD:**

openai api completions.create -m <FINE\_TUNED\_MODEL> -p <YOUR\_PROMPT>

example: openai api completions.create -m new\_name -p ‘’ Prestação de serviços. Coletaresiduos. Meioambiente. Residuossolidos. ->”

1. **Node.js:**

const response = await openai.createCompletionFromModel({

model: FINE\_TUNED\_MODEL

prompt: YOUR\_PROMPT,

});

1. **Python:**

import openai

openai.Completion.create(

model=FINE\_TUNED\_MODEL,

prompt=YOUR\_PROMPT)

I can also develop a complete step by step video for you to understand better.

1. Other GPT-3 Parameters:

To get best results with parameters such as temperature, we should then input different values to get the best for our applications. This is also available in the playground but without the fine tuning of course.

The role, significance and usage of these parameters will be discussed in the next blog.

AI Studio offers B2B software services related to field of AI. It is based on a team of Machine Learning Consultants having expertise in NLP, Computer Vision, Data Science and Web Development.

**References:**

[1] <https://openai.com/charter/>

[2] <https://beta.openai.com/playground>

[3] <https://beta.openai.com/docs/guides/search>

[4] <https://beta.openai.com/docs/guides/fine-tuning>

[5] <https://beta.openai.com/docs/guides/classifications>

[6] <https://beta.openai.com/docs/guides/answers>