Falcon AI: Open Source Large Language Model

ARTIFICIAL INTELLIGENCE BEGINNER GENERATIVE AI PYTHON PYTORCH

Introduction

Ever since the launch of GPT (Generative Pre Trained) by Open AI, the world has been taken by storm by Generative AI. From that period on, many Generative Models have come into the picture. With each release of new Generative Large Language Models, AI kept on coming closer to Human Intelligence. However, the Open AI community made the GPT family of powerful Large Language Models closed source. Fortunately, Falcon AI, a highly capable Generative Model, surpassing many other LLMs, and it is now open source, available for anyone to use. Falcon AI integrates cutting-edge machine learning techniques, offering users unprecedented capabilities in generating natural language text.

Learning Objectives

- To understand why Falcon AI topped the LLM Leaderboard
- To learn the capabilities of Falcon AI
- Observing the Falcon AI Performance
- Setting up Falcon AI in Python
- Testing Falcon AI in LangChain with custom Prompts

This article was published as a part of the Data Science Blogathon.

Table of contents

What is Falcon AI?

Falcon AI, mainly Falcon LLM 40B, is a Large Language Model released by the UAE's Technology Innovation Institute (TII). The 40B indicates the 40 Billion parameters used by this Large Language Model uses. The TII has even developed a 7B, i.e., 7 billion parameters model that's trained on 1500 billion tokens. In comparison, the Falcon LLM 40B model is trained on 1 trillion tokens of RefinedWeb. What makes this LLM different from others is that this model is transparent and Open Source.

The Falcon, an autoregressive decoder-only model, represents a significant advancement in AI models. Its development involved rigorous training on the AWS Cloud for a continuous period of two months, utilizing 384 GPUs. The pretraining data predominantly comprised publicly available sources, supplemented by select data extracted from research papers and discussions on social media platforms.

Why Falcon AI?

Large Language Models are profoundly influenced by the data they're trained on, and their responsiveness fluctuates with evolving datasets. For Falcon, we meticulously curated the training data, drawing from a diverse array of sources including extracts from high-quality websites (RefinedWeb Dataset). Through meticulous filtering and de-duplication processes, we ensured the integrity and richness of the dataset, setting a solid foundation for Falcon's training. In addition to utilizing readily available data sources, we tailored the dataset to suit Falcon's architecture, optimizing it for inference tasks. As a result, Falcon exhibits superior performance compared to state-of-the-art models such as Google, Anthropic, Deepmind, LLaMa, and others, as evidenced by its top ranking on the OpenLLM Leaderboard. This demonstrates Falcon's prowess in handling diverse AI models and datasets.

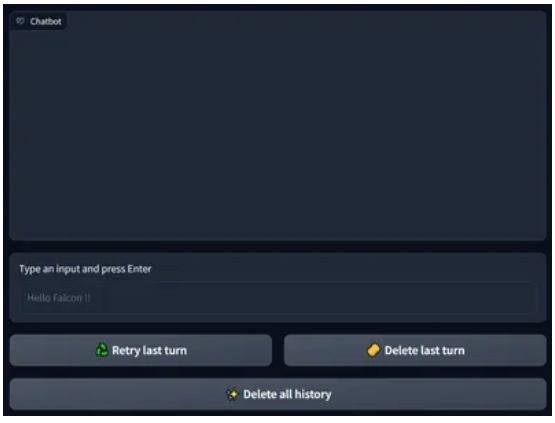
Apart from all this, the main differentiator is that it's open-sourced, thus allowing for commercial use with no restrictions. So anyone can finetune Falcon with their data to create their application from this Large Language Model. Falcon even comes with specialized versions known as Falcon-7B-Instruct and Falcon-40B-Instruct, which are pre-trained on conversational data. These can be seamlessly integrated to develop chat applications tailored to specific needs, leveraging the power of artificial intelligence.

First Look: Falcon Large Language Model

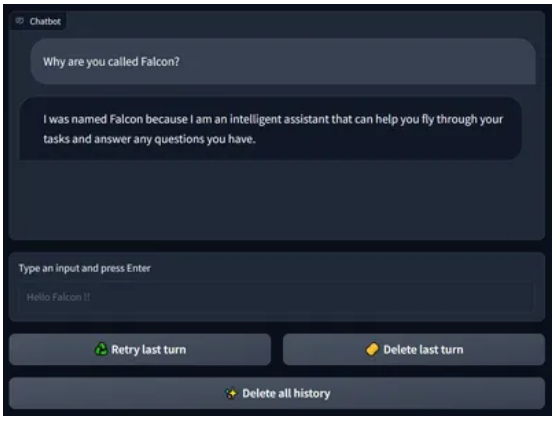
In this section, we will be exploring one of Falcon's models, and the variant we'll delve into is the Falcon-40B Model, known for its remarkable performance on the OpenLLM Leaderboard charts. However, let's not overlook the Falcon 180B, a model that also holds its own in various contexts.

For our demonstration, we'll focus on the Falcon-40B-Instruct version, meticulously fine-tuned on conversational data to ensure seamless interaction. To engage with the Falcon Instruct model, one convenient avenue is through HuggingFace Spaces. Notably, HuggingFace offers a dedicated Space for the Falcon-40B-Instruct Model, aptly named the Falcon-Chat demo.

Click here to explore the site and experience its capabilities firsthand.



After opening the site, scroll down to see the chat section, which is similar to the pic above. In the “Type an input and press Enter” field, enter the query you want to ask the Falcon Model and press Enter to start the conversation. Let’s ask a question to the Falcon Model and see its output.



In Image 1, we can see the response generated. That was a good response from the Falcon-40B model to the query. We have seen the working of Falcon-40B-Instruct in the HuggingFace Spaces. But what if we want to work with it in a specific code? We can do this by using the Transformers library. We will go through the necessary steps now.

Download the Packages

```
!pip install transformers accelerate einops xformers
```

We install the transformers package to download and work with the state-of-the-art models that are pre-train, like the Falcon. The accelerate package enables us to run PyTorch models on whichever system we are working with, and currently, we are using Google Colab. The einops and xformers are the other packages that support the Falcon model.

Import Libraries

Now we need to import these libraries to download and start working with the Falcon model. The code will be:

```
from transformers import AutoTokenizer, AutoModelForCausalLM import transformers import torch model = "tiiuae/falcon-7b-instruct" tokenizer = AutoTokenizer.from_pretrained(model) pipeline = transformers.pipeline( "text-generation", model=model, tokenizer=tokenizer, torch_dtype=torch.bfloat16, trust_remote_code=True, device_map="auto", max_length=200, do_sample=True, top_k=10, num_return_sequences=1, eos_token_id=tokenizer.eos_token_id )
```

Steps Involved in Testing Falcon 7B

- Firstly, we need to provide the path to the model that we will be testing. Here we will be working with the Falcon-7B-Instruct model because it takes less space in GPU and can be can with the free tier in the Google Colab.
- The Falcon-7B-Instruct Large Language Model link is stored in the model variable.

- To download the tokenizer for this model, we write the from_pretrained() method from the AutoTokenizer class present in transformers.
- To this, we provide the LLM path, which then downloads the Tokenizer that works for this model.
- Now we create a pipeline. When creating the pipelines, we provide the necessary options, like the model we are working with and the type of model, i.e., “text-generation” for our use case.
- The type of tokenizer and other parameters are provided to the pipeline object.

Let’s try observing Falcon’s 7B instruct model output by providing the model with a query. To test the Falcon model, we will write the below code.

```
sequences = pipeline( "Create a list of 3 important things to reduce global warming" ) for seq in sequences: print(f"Result: {seq['generated_text']}")
```

We asked the Falcon Large Language Model to list the three important things to reduce global warming. Let’s see the output generated by this model.

```
Result: Create a list of 3 important things to reduce global warming
1. Reduce carbon emissions by using renewable energy sources.
2. Promote the use of public transport and carpooling to limit the number of private vehicles on the road.
3. Conserve and preserve resources by recycling and reducing water consumption.
```

We can see that the Falcon 7B Instruct model has produced a good result. It pointed out the root problems for the cause of global warming and even provided the appropriate solution for tackling the issues, thus reducing global warming.

Falcon AI with LangChain

Langchain is a Python Library designed to facilitate the development of applications leveraging Large Language Models. One of its notable features is the HuggingFacePipeline, tailored for models hosted within the HuggingFace ecosystem. With this capability, integrating Falcon with LangChain becomes a feasible option, enabling seamless utilization of AI models in your applications.

Install LangChain Package

```
!pip install langchain
```

This will download the latest langchain package. Now, we need to create a Pipeline for the Falcon model, which we will do so by:

```
from langchain import HuggingFacePipeline llm = HuggingFacePipeline(pipeline = pipeline, model_kwargs = {'temperature':0})
```

- We call the HuggingFacePipeline() object and pass the pipeline and the model parameters.
- Here we are using the pipeline from the “First Look: Falcon Large Language Model” section.
- For the model parameters, we are providing the temperature a value of 0, which makes the model not hallucinate much(creating its own answers).
- All this, we pass to a variable called **llm**, which stores our Large Language Model.

Incorporating API

Now we know that LangChain contains PromptTemplate, which allows us to alter the answers produced by the Large Language Model. Additionally, we have LLMChain, which seamlessly integrates the PromptTemplate and the LLM together. Let's write code utilizing these methods, incorporating an API for enhanced functionality.

```
from langchain import PromptTemplate, LLMChain template = """ You are a intelligent chatbot. You reply should be in a funny way. Question: {query} Answer:""" prompt = PromptTemplate(template=template, input_variables=["query"]) llm_chain = LLMChain(prompt=prompt, llm=llm)
```

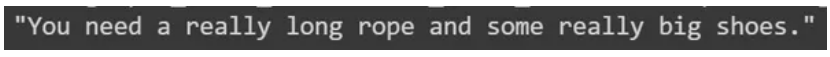
Steps Involved

- Firstly, we define a template for the Prompt. The template describes how our LLM should behave, that is, how it should answer the questions given by the user.
- This is then passed to the PromptTemplate() method and stored in a variable
- Now we need to chain the Large Language Model and the Prompt together, which we do so by providing them to the LLMChain() method.

Now our model is ready. According to the Prompt, the model must funnily answer a given question. Let's try this with an example code.

```
query = "How to reach the moon?" print(llm_chain.run(query))
```

So we gave the query "How to reach the moon?" to the model. The answer is below:



The response generated by the Falcon-7B-Instruct model is indeed funny. It followed the prompt given by us and generated the appropriate answer to the given question. This is just one of the few things we can discover with this new Open Source Model.

Conclusion

In this article, we have discussed a new Large Language Model called Falcon. Falcon AI has taken the top spot on the OpenLLM Leaderboard by beating top models like Llama, MPT, StableLM, and many more. The best thing about this Model is that it's Open Source, meaning that anyone can develop applications with Falcon AI for commercial purposes.

Key Takeaways

- Falcon-40B is right now, positioned at the top of the OpenLLM Leaderboard
- Flacon has open-sourced both the 40 Billion and the 7 Billion models

- You can work with the Instruct models of Falcon, which are pre-trained on conversations, to quickly get started.
- Optimise Falcon's architecture for Inference.
- Finetune this model to build different applications.

Frequently Asked Questions

Q1. What is Falcon LLM good for?

A. Falcon LLM, favored by Microsoft and NVIDIA, is great for text tasks like generation and understanding. It's handy for customer support, content creation, and sentiment analysis, thanks to its human-like text skills. Integration with platforms like HuggingFace Spaces makes it even more accessible.

Q2. What is Falcon model used for?

A. The Falcon model, an open-source AI, is used for natural language processing tasks and dives into conversational AI, enabling developers to create chatbots and virtual assistants.

Q3. How good is the Falcon-40B model?

A. Falcon-40B has topped the chart in the OpenLLM Leaderboard. It has surpassed state-of-the-art models like Llama, MPT, StableLM, and many more. The Falcon has an optimized architecture for inference tasks.

Q4. Is Falcon better than ChatGPT?

A. Determining if Falcon is better than ChatGPT in NLP depends on factors like task requirements and model capabilities. Falcon, especially Falcon-40B, excels in certain tasks, but ChatGPT also offers strengths across various NLP applications. It's essential to evaluate both based on specific needs and task nuances.

Q5. What does Falcon AI do?

A. Falcon AI specializes in natural language processing, which involves teaching computers to understand and interact with human language. They create solutions like sentiment analysis, chatbots, and language translation to help businesses analyze data and improve communication.

Article Url - https://www.analyticsvidhya.com/blog/2023/07/falcon-ai-the-new-open-source-large-language-model/

Ajay Kumar Reddy