

POC:- Proof of Concept

Research Book Name Generation

"To generate Research book name by giving prompts to Model"

Proposed Methodology:

Transformers used : LaMini-Flan-T5-77M, LaMini-Flan-T5-783M, LaMini-Flan-T5-248M
Imported Transformers from Hugging face AI

Used GPT2LMHeadModel, GPT2Tokenizer

GPT2LMHeadModel:

GPT-2 is a powerful language model that belongs to the family of transformer-based models. It is capable of generating coherent and contextually relevant text based on the input it receives. "LMHead" stands for "Language Model Head," and it refers to the fact that the model is designed for language modeling tasks.

The GPT2LMHeadModel class from the transformers library is used to load and interact with the pre-trained GPT-2 model. This class allows us to generate text, perform language modeling tasks

GPT2Tokenizer:

Tokenization is the process of breaking down a sequence of text into smaller units called tokens. The GPT2Tokenizer is specifically designed to tokenize text in a way that is compatible with the GPT-2 model. It handles the conversion of text into numerical representations (token IDs) that the model can understand.

The GPT2Tokenizer class is used to tokenize input text before feeding it into the GPT-2 model and to decode the model's generated output back into text. It helps maintain consistency between the tokenization process during training and inference.

Analysis:

This task is to generate text-to-text. so,

Firstly I explored the AI tool to generate a Research Book Name and found one tool named Writer Buddy AI:

<https://writerbuddy.ai/free-tools/book-title-generator>

Result of this tool:

Input: Sibling Murder

Output: "The Sinister Brotherhood: A Tale of Sibling Murder"

This can be used to generate a research book name generation but our task is to find free alternative models so I have explored many open-source transformers on Hugging Face AI and after doing some research I found some transformers like **LaMini-Flan-T5-77M**, **LaMini-Flan-T5-783M**, **LaMini-Flan-T5-248M**

Result

Using LaMini-Flan-T5-248M :

Query: Sibling Murder

Research Book Name: "Siblings Murder" by Rebecca Skloot.

Query: Murder

Research Book Name: "The Murder of the World" by Robert Frost.

Query: Cheque Bounce

Research Book Name: "The Cheque Bounce" by John Grisham

Using LaMini-Flan-T5-77M:

Query: Sibling Murder

Research Book Name: "The Case of the Murder"

Using LaMini-Flan-T5-783M:

Query: Sibling Murder

Research Book Name: ""Siblings Murder: A Case Study""

Query: Is a last will considered the final will of a person?

Research Book Name: ""Last Wills: A Comprehensive Guide to Wills and Estate Planning""

References:

<https://huggingface.co/models?other=instruction+fine-tuning>

<https://huggingface.co/tasks/text-generation>

https://youtu.be/Kilc1mEHInY?si=QP3kOqf8uAv-_zMV

Google Bard

Chat GPT

Generate Enhance Facts

"To generate enhance facts from the given sentence"

Proposed Methodology:

For the generation of enhanced facts at the beginning I was using a Natural Language tool kit like tokenization and lemmatizer by which I was able to get the root word and then I can modify it to its similar meaning but that was far away from the expected result because we have to enhance the fact not to paraphrase the sentence:

Here is the code snippet to get the root word of any word:

```
word = "swimming"
```

```
lemma = lemmatizer.lemmatize(word, pos='v') # 'v' indicates verb
print(f"Original Word: {word}, Lemma: {lemma}")
```

Used PorterStemmer to stem the word

```
porter_stemmer=PorterStemmer()
words=['Amazing', 'Happily', 'Playing', 'Singing']
stem_word=[porter_stemmer.stem(word) for word in words]
print("Original word", words)
print("Stemmed word", stem_word)
```

Original word ['Amazing', 'Happily', 'Playing', 'Singing']

Stemmed word ['amaz', 'happili', 'play', 'sing']

Used WordNetLemmatizer to generate synonyms of the word:

```
from nltk.stem import WordNetLemmatizer

# Create a WordNetLemmatizer object
lemmatizer = WordNetLemmatizer()

# Example usage
word = "better"
lemma = lemmatizer.lemmatize(word, pos='a')
print(f"Original Word: {word}, Lemma: {lemma}")
```

Original Word: better, Lemma: good

But this way I was only able to paraphrase the sentence, so I got to understand that either we have to give internal knowledge embedded in the input text or we have to use external knowledge from outside sources such as knowledge base and knowledge graph in the text generation system

Analysis:

I have researched for AI tools that can perform the same task other than Open AI and Google Bard for free, I have encountered one tool that was free but for limited access Tool is:

<https://deepai.org/chat/text-generator>

Prompt:- Loan taken from Bank

Response:-

A loan obtained from a financial institution, such as a bank, is a financial product that provides an individual or business with access to funds for various purposes. This type of loan typically involves repayment of principal and interest over an agreed-upon period, with interest rates and fees varying depending on the loan terms and the borrower's creditworthiness. Loans from banks can be used for various purposes, such as purchasing a home, starting a business, or covering unexpected expenses.

Result

By Using Lamini-Flan-T5-248M

For text to text generation

```
classifier  
=pipeline(task="text2text-generation",model="MBZUAI/LaMini-Flan-T5-248M")
```

Fact: Loan pending with bank

[{'generated_text': 'A Loan pending with bank'}]

Fact: Physical or Mental Torture

[{'generated_text': 'Physical Torture: a physical discomfort or discomfort caused by physical or mental stress. Mental Tor'}]

Fact: "The accused did not create or submit forged documents to the authority."

[{'generated_text': 'The accused did not submit forged documents to the authority.'}]

References:

<https://deepai.org/chat/text-generator>

<https://huggingface.co/models?other=instruction+fine-tuning>

<https://huggingface.co/tasks/text-generation>

https://youtu.be/CDmPBsZ09wg?si=IAvuFyD0tz_1Om8E

Google Bard

Chat GPT

Thank You

Documented By:- Aditya Raj