**Next Sentence Prediction with BERT and Wikipedia Sentence Matching**

**Overview**

This Python script leverages the BERT (Bidirectional Encoder Representations from Transformers) model for next sentence prediction and sentence matching using Wikipedia content. It predicts the likelihood of a user-provided sentence being a continuation of sentences extracted from a specified Wikipedia page.

**Dependencies**

- [transformers](https://github.com/huggingface/transformers) library by Hugging Face

- [torch](https://pytorch.org/) library

- [nltk](https://www.nltk.org/) library for natural language processing

- [wikipedia-api](https://pypi.org/project/Wikipedia-API/) library for accessing Wikipedia content

Ensure that the necessary dependencies are installed using:

```bash

pip install transformers torch nltk wikipedia-api

```

**Execution**

1. Clone the repository and navigate to the project folder:

```bash

git clone https://github.com/your-username/wikipedia-sentence-matching.git

cd wikipedia-sentence-matching

```

2. Install dependencies:

```bash

pip install transformers torch nltk wikipedia-api

```

3. Run the script:

```bash

python sentence\_matching\_with\_wikipedia.py

```

4. Follow the prompts to input a sentence and the topic you're talking about. The script will fetch content from Wikipedia, tokenize it into sentences, and identify the most suitable sentence based on BERT predictions.

**Code Structure**

- The script uses BERT for next sentence prediction and sentence matching with Wikipedia content.

- The `predict\_next\_sentence` function takes two input sentences and predicts the probability of the second sentence being a continuation of the first using the BERT model.

- Wikipedia content is fetched based on user input, and sentences are tokenized for analysis.

**Example**

```python

input\_sentence = input('Type a sentence:\n')

input\_topic = input("Type the topic you're talking about: ")

wikipedia\_content = wikipedia.page(input\_topic).content

sentences = nltk.tokenize.sent\_tokenize(wikipedia\_content)

best\_match\_sentence = find\_best\_matching\_sentence(input\_sentence, sentences)

print(f"Best suited sentence from the Wikipedia content: {best\_match\_sentence}")

**Usage Tips**

1. Explore different topics on Wikipedia to observe BERT's predictions.

2. Customize the script for specific use cases by adjusting the input prompts and Wikipedia content retrieval.

**Timeline:**

13-02-24 ​

* Look for requirements​.
* Learn about Transformers and BERT model​
* Search for codes Online and see what matches the best for the model​
* Obtain the ideal dataset that can be used.​

14-02-24​

* Exploratory data analysis​
* Dataset preprocessing​
* Make a Flow chart of a working model with BERT

15-02-24​

* Make a working base model with BERT​.
* Analyse the optimum evaluation metrics that should be used ​

16-02-24​

* Optimise the model to the give the best result​
* Evaluation on test set and record the result​

19-02-24​

* Model Submission​