OdiaQABot

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology

(Deemed to be University), Patiala – 147004

Natural Language Processing Project

Submitted By:

Jasmine Das (102117016,3CS1)

Simran Minhas (102117020,3CS1)

Hansika Sachdeva (102103309,3CO11)

Submitted To:

Dr. Jasmeet Singh

NLP Application

OdiaQABot is a Natural Language Processing (NLP) application designed to facilitate **Question Answering** in the **Odia** language. It aims to bridge language barriers and enhance accessibility to information for Odia speakers by providing a user-friendly interface for querying and retrieving relevant answers in their native language.

Question Answering is a computer science discipline within the fields of information retrieval and natural language processing, which focuses on building systems that automatically answer questions posed by humans in a natural language. A computer understanding of natural language consists of the capability of a program system to translate sentences into an internal representation so that this system generates valid answers to questions asked by an user. Valid answers mean answers relevant to the questions posed by the user. As the internal representation of natural language, sentences must adequately map semantics of this statement, the most natural approach is in the simulation of facts contained in the sentences using a description of real objects as well as actions and events connected with these objects. To form an answer it is necessary, in the first place, to execute the syntax and semantic analysis of a question.

With the help of OdiaQABot users can input questions directly in Odia, and it utilizes **Google's Flan T5** machine learning model for processing. Flan T5 is a state-of-the-art model trained specifically for text-to-text tasks, making it ideal for question answering in Odia.

Upon receiving a user query, OdiaQABot employs Flan T5 to analyze the input text and generate accurate responses. The model's advanced natural language understanding capabilities enable it to comprehend the nuances of Odia text and extract relevant information from a knowledge base or dataset.

The question answering process is seamlessly integrated into the user interface, providing a streamlined experience for Odia speakers seeking information. Users receive answers directly in Odia, eliminating the need for translation and ensuring clarity and comprehension.

OdiaQABot's architecture leverages modern NLP frameworks and technologies to deliver fast and reliable question answering capabilities in the Odia language. Implementation details include the use of Python programming language and Flax library for deploying and fine-tuning the Flan T5 model.

Dataset

The dataset for our project is collected from the **hugging face, aibharat-IndicQA**.The Indic QA dataset is designed for question answering tasks, with a focus on Indic languages. It contains questions paired with corresponding contexts and answers. The dataset aims to facilitate research and development in question answering systems for Indic languages. This dataset has total ten languages that are **Assamese (as),Bengali (bn),Hindi (hi),Kannada (kn),Marathi (mr),Malayalam (ml),Punjabi (pa),Oriya (or),Tamil (ta),Telugu (te)**.The ai4bharat /IndicQA dataset provides a valuable resource for training your Odia question answering bot. Since it includes Odia (Oriya) as one of the ten languages, we have specifically accessed the Odia portion of the dataset. This dataset contains question-answer pairs along with corresponding contexts, which is ideal for training machine learning models for question answering tasks.

Features of the dataset:

1.ID: Identifier for each data instance.

2.Context: The passage or context providing information relevant to answering the question.

3.Question: The question posed by the user.

4.Answers: The possible answers to the question, provided as a sequence.

5.Number of rows:1680

Preprocessing of data:

1.Training Data: Provides question-answer pairs with contexts in Odia, ideal for training machine learning models for Odia QA.

2.Focus on Odia: Enables training specifically on Odia data, improving the bot's accuracy in answering Odia user queries.

3.Data Exploration: Analyzing the size, quality (format, consistency), and domain coverage of the Odia data subset before training.

4.Data Preprocessing: Clean text data, handle inconsistencies, and potentially apply Odia-specific tokenization techniques.

5.Model Selection and Training: Choose a suitable QA model architecture (e.g., transformers) based on data size and complexity. Training the model on the preprocessed Odia data.

LINK- https://huggingface.co/datasets/ai4bharat/IndicQA

Transformer Model

FLAN T5

Google's Flan-T5 transformer model, released in October 2022, is an advanced variant of the original T5 (Text-to-Text Transfer Transformer) model. Flan-T5 builds upon the architecture of T5, which is based on the encoder-decoder structure commonly used in transformer models. The encoder processes the input sequence to generate context-aware representations of each token, while the decoder uses these representations to generate the output sequence token by token.

T5 is a text-to-text transfer model, which can be fine-tuned to perform a wide range of natural language understanding tasks, such as text classification, language translation, and question-answering. It's trained on a massive amount of text data, which allows it to understand and generate a wide range of natural language. T5 introduced the "prefix" approach to transfer learning, where the model is fine-tuned for a specific task by training it with a prefix added to the input text. Flan-T5 further improved over the regular T5 model by fine-tuning on a more extensive and varied set of tasks. This fine-tuning process, known as "instruction tuning," makes Flan-T5 Base highly capable of handling complex natural language processing tasks. Compared to its predecessors, FLAN-T5 is more efficient in terms of computational resources and inference time, and it demonstrates higher accuracy in tasks requiring comprehension and reasoning.

Flan-T5 is available in several versions, including Flan-T5 Small, Base, Large, XL, and XXL with the total number of parameters ranging from 80 million to 11 billion. In this project, we used the Flan-T5 base model and fine-tuned it on IndicQA dataset to perform question answering task in Odia language. The Flan-T5 base model contains 12 layers (or hops) for both the encoder and decoder with 12 attention heads in each encode and decoder and approximately 220 million parameters in total.

The Flan-T5 model first splits the input question and context into tokens. These token embeddings are processed through the encoder, which uses multi-head self-attention mechanisms and feed-forward networks to capture the contextual relationships among tokens, resulting in output embeddings of 768 dimensions that represent the input text in a high-dimensional space. The self-attention mechanism allows the model to understand the relationships and importance of different tokens within the sequence. Thus, the resulting embeddings encapsulate the semantic information of the entire input. The decoder then takes these contextual embeddings and, using its own self-attention

mechanisms, generates the answer token by token. It starts by applying self-attention to the tokens generated so far and incorporates contextual information from the encoder's output through encoder-decoder attention mechanisms. Each generated token is influenced by the previously generated tokens and the contextual embeddings, ensuring the generated answer is coherent and contextually accurate.

Result

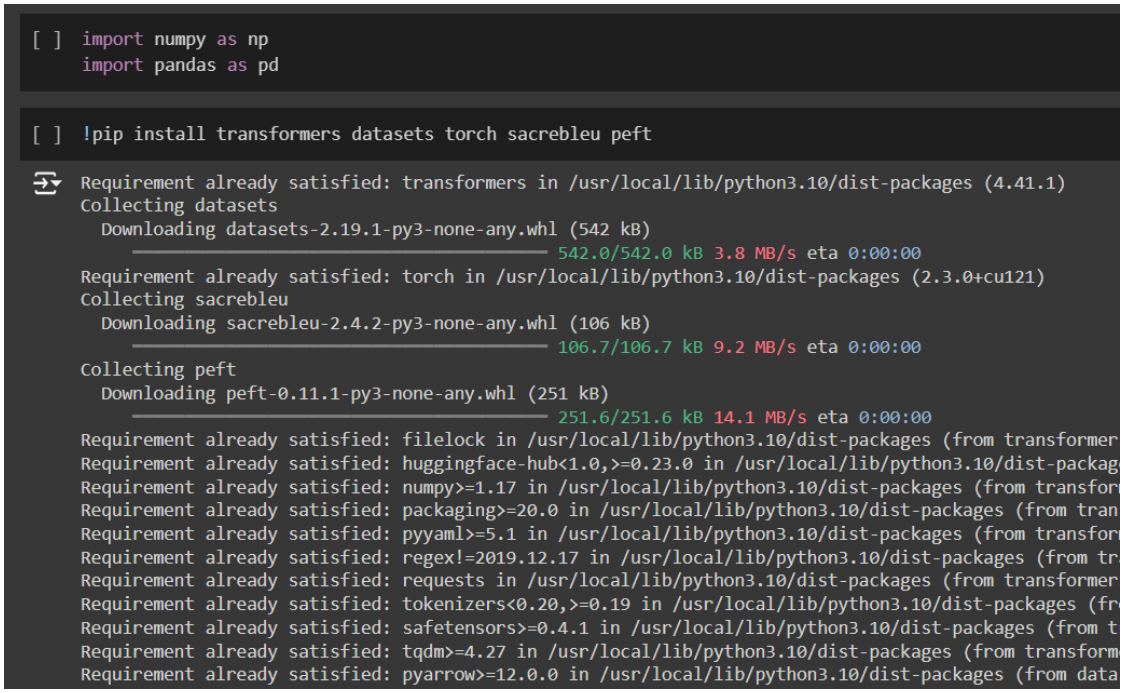
Accuracy:

- **BLEU SCORE:** 0.4050021510445334
- **F1 SCORE:** 0.5322555704901326

Python Code(Link & Screenshots)

Link to Notebook:

https://colab.research.google.com/drive/1qUxIj-XZMNwTeZVaIWYMvMKOI8J8ruPB?usp=sharing



```
[ ] import numpy as np
    import pandas as pd

[ ] !pip install transformers datasets torch sacrebleu peft

    Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-packages (4.41.1)
    Collecting datasets
      Downloading datasets-2.19.1-py3-none-any.whl (542 kB)
         ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 542.0/542.0 kB 3.8 MB/s eta 0:00:00
    Requirement already satisfied: torch in /usr/local/lib/python3.10/dist-packages (2.3.0+cu121)
    Collecting sacrebleu
      Downloading sacrebleu-2.4.2-py3-none-any.whl (106 kB)
         ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 106.7/106.7 kB 9.2 MB/s eta 0:00:00
    Collecting peft
      Downloading peft-0.11.1-py3-none-any.whl (251 kB)
         ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 251.6/251.6 kB 14.1 MB/s eta 0:00:00
    Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from transformer
    Requirement already satisfied: huggingface-hub<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packag
    Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from transfor
    Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from tran
    Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transfor
    Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from tr
    Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformer
    Requirement already satisfied: tokenizers<0.20,>=0.19 in /usr/local/lib/python3.10/dist-packages (fr
    Requirement already satisfied: safetensors>=0.4.1 in /usr/local/lib/python3.10/dist-packages (from t
    Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transform
    Requirement already satisfied: pyarrow>=12.0.0 in /usr/local/lib/python3.10/dist-packages (from data
```

Preprocessing Datasets

```
[ ] #training for question answering
    from datasets import load_dataset
    dataset = load_dataset("ai4bharat/IndicQA", 'indicqa.or')
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:89: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/datasets/load.py:1486: FutureWarning: The repository for ai4bha
You can avoid this message in future by passing the argument `trust_remote_code=True`.
Passing `trust_remote_code=True` will be mandatory to load this dataset from the next major release of
  warnings.warn(
Downloading builder script: 100% 4.47k/4.47k [00:00<00:00, 51.6kB/s]
Downloading readme: 100% 1.26k/1.26k [00:00<00:00, 9.62kB/s]
Downloading data: 100% 2.07M/2.07M [00:00<00:00, 8.81MB/s]
Downloading data: 100% 2.59M/2.59M [00:00<00:00, 11.6MB/s]
Downloading data: 100% 1.90M/1.90M [00:00<00:00, 1.32MB/s]
Downloading data: 100% 2.67M/2.67M [00:00<00:00, 9.75MB/s]
Downloading data: 100% 1.57M/1.57M [00:00<00:00, 5.26MB/s]
Downloading data: 100% 2.82M/2.82M [00:00<00:00, 11.3MB/s]
Downloading data: 100% 2.18M/2.18M [00:00<00:00, 9.98MB/s]
```

```
for example in dataset['test']:
    sequence_values = example['question']
    print(sequence_values)
```

```
କେଉଁ ସରକାର ରିପୋର୍ଟ କରିଛନ୍ତି ଯେ ଆରଏସଏସ କୌଣସି ସରକାରୀ ଆଦେଶକୁ ଉଲ୍ଲଂଘନ କରି ନାହାନ୍ତି?
କେଉଁ ବର୍ଷରେ ଆରଏସଏସ ନେତାମାନେ ସମସ୍ତ କାର୍ଯ୍ୟକଳାପରୁ ଦୂରେଇ ରହିଥିଲେ?
ଆରଏସଏସର କେଉଁ ନେତା କହିଥିଲେ ଯେ ଆରଏସଏସ ଭାରତ ଛାଡିବା ଆନ୍ଦୋଳନକୁ ସମର୍ଥନ କରୁନାହିଁ?
କାର୍ଯ୍ୟକଳାପରୁ ନିବୃତ୍ତ ହେବାକୁ ଆରଏସଏସ ନେତାମାନଙ୍କୁ କିଏ ନିର୍ଦ୍ଦେଶ ଦେଇଛନ୍ତି?
ବ୍ରିଟିଶ ସରକାର ଦର୍ଶାଇଛନ୍ତି ଯେ ଆରଏସଏସ କେଉଁମାନଙ୍କ ବିରୁଦ୍ଧରେ କୌଣସି ନାଗରିକ ଅବମାନନାକୁ ସମର୍ଥନ କରୁନାହାନ୍ତି?
ମିସନାରୀମାନେ କେଉଁ ଶ୍ରୀଷ୍ଟାବ୍ଦରେ ଓଡ଼ିଶାରେ ପ୍ରଥମ ଇଂରାଜୀ ୪ଟି ସ୍କୁଲ ପ୍ରତିଷ୍ଠା କରିଥିଲେ?
କେଉଁ  ବିଶ୍ୱବିଦ୍ୟାଳୟ ଶିକ୍ଷାର ଏକ ପ୍ରମୁଖ କେନ୍ଦ୍ର ଅଟେ ?
କେତେ ମସିହାରେ ପୁରୀ ଇଂରାଜୀ ବିଦ୍ୟାଳୟ ପ୍ରତିଷ୍ଠା ହେଲା?
ପ୍ରଥମେ ଏଠାରେ କଣ ଖୋଲା ହେଲା ?
୧୮୪୧ ମସିହାରେ ଆଉ କେତେଟି ସ୍ଥାନରେ ଇଂରାଜୀ ବିଦ୍ୟାଳୟ ପ୍ରତିଷ୍ଠା ହେଲା?
ପରବର୍ତ୍ତୀ ସମୟରେ ଏହି ସ୍କୁଲର ପରିଚାଳନା ଦାୟିତ୍ୱ କିଏ ନେଲେ ?
କେତେ ମସିହାରେ କଟକଠାରେ ଆଧୁନିକ ବିଦ୍ୟାଳୟ ପ୍ରତିଷ୍ଠା ହେଲା?
covid-19 ପାଇଁ ଓଡିଶାର କିଛି ଅଂଶ କେତେ ଘଣ୍ଟା ପାଇଁ ସମ୍ପୂର୍ଣ୍ଣ ବନ୍ଦ ହୋଇଗଲା?
ଏପ୍ରିଲରେ ଓଡିଶାରେ କେତେ ଲୋକ କୋଭିଡ୍ ପ୍ରଭାବିତ ହୋଇଥିଲେ?
ପ୍ରଥମ COV-2 କେବେ ଚିହ୍ନଟ ହେଲା?
ପ୍ରଥମ କୋଭିଡ୍ ସଂକ୍ରମିତ ବ୍ୟକ୍ତି ଓଡିଶା ଫୁଆଡେ ଫେରିଗଲେ?
ଓଡିଶାରେ କୋଭିଡ୍ -19 ରେସପନ୍ସ କମିଟିର ମୁଖ୍ୟ ଭାବରେ କିଏ ନିଯୁକ୍ତ ହେଲା?
ଅଂକୋର୍ ୱାଟ୍ କେଉଁ ଦିଗକୁ ମୁହଁ କରି ଦଣ୍ଡାୟମାନ?
କେଉଁମାନଙ୍କ କ୍ଷେତ୍ରରେ ଶବଦାହ ସମୟରେ ସମସ୍ତ ବିଧି ବିପରୀତ ଦିଗରେ ପରିଚାଳିତ ହୁଏ?
କେଉଁ କେଉଁ ଗବେଷକମାନେ ଅନୁମାନ କରନ୍ତି ଯେ ରାଜା ସୂର୍ଯ୍ୟବର୍ମନ ବୋଧହୁଏ ଅଂକୋର୍ ୱାଟ ମନ୍ଦିରକୁ ନିଜର ସମାଧୀ ମନ୍ଦିର ଭାବେ ନିର୍ମାଣ କରି
ମୌରିସ୍ ଗ୍ଲେଜ୍ ଓ ଜର୍ଜ୍ କୋଡେ ପରି ଅନେକ ଗବେଷକମାନେ ଅନୁମାନ ଅନୁସାରେ କେଉଁ ରାଜା ବୋଧହୁଏ ଅଙ୍କୋର୍ ୱାଟ ମନ୍ଦିରକୁ ନିଜର ସମାଧୀ ସ
ଅମିତାଭଙ୍କ ବାପା ଜଣେ କ'ଣ ଥିଲେ ?
ଇନ୍‌କିଲାବ୍‌ର ଅର୍ଥ କ'ଣ ?
କିଏ ନାମ ପରିବର୍ତ୍ତନ କରିଥିଲେ ?
ଅମିତାଭଙ୍କ ପୂର୍ବ ପୁରୁଷ ଉତ୍ତରପ୍ରଦେଶର କେଉଁ ଗାଁରେ ରହୁଥିଲେ ?
କେଉଁ ସହରରେ ଅମିତାଭ ବଚ୍ଚନଙ୍କ ଜନ୍ମଗ୍ରହଣ ?
ଅମିତାଭଙ୍କ ମା' କେଉଁ ସଂପ୍ରଦାୟର ଥିଲେ ?
ଅମିତାଭ ବଚ୍ଚନ କେତେ ମସିହାରେ ଜନ୍ମଗ୍ରହଣ କରିଥିଲେ ?
କେଉଁ ଶିକ୍ଷାନୁଷ୍ଠାନରୁ ଅମିତାଭ ବଚ୍ଚନ ସ୍ନାତକ ଶିକ୍ଷାପ୍ରାପ୍ତ ହୋଇଥିଲେ ?
```

```
[ ] for example in dataset['test']:
        sequence_values = example['answers']
        print(sequence_values)
```

```
{'text': ['ବ୍ରିଟିଶ ସରକାର'], 'answer_start': [717]}
{'text': ['୧୯୪୨ ମସିହାରେ'], 'answer_start': [413]}
{'text': ['ଏମ.ଏସ. ଗୋଲୱାଲକର'], 'answer_start': [169]}
{'text': ['ଏମ.ଏସ. ଗୋଲୱାଲକର'], 'answer_start': [169]}
{'text': ['ବ୍ରିଟିଶ ସରକାର'], 'answer_start': [718]}
{'text': [''], 'answer_start': [None]}
{'text': [''], 'answer_start': [None]}
{'text': ['.୧୮୩୫'], 'answer_start': [260]}
{'text': ['ଆଧୁନିକ ସ୍କୁଲ'], 'answer_start': [177]}
{'text': ['୮ଟି'], 'answer_start': [335]}
{'text': ['କମ୍ପାନୀ ସରକାର'], 'answer_start': [732]}
{'text': ['ଖ୍ରୀ.୧୮୨୨'], 'answer_start': [190]}
{'text': [''], 'answer_start': [None]}
{'text': [''], 'answer_start': [None]}
{'text': [''], 'answer_start': [None]}
{'text': [''], 'answer_start': [None]}
{'text': [''], 'answer_start': [None]}
{'text': ['ପଶ୍ଚିମ'], 'answer_start': [144]}
{'text': ['ବ୍ରାହ୍ମଣମାନଙ୍କ'], 'answer_start': [446]}
{'text': ['ମୌରିସ୍ ଗ୍ଲେଜ୍ ଓ ଜର୍ଜ୍ କୋଡେ'], 'answer_start': [196]}
{'text': ['ସୂର୍ଯ୍ୟବର୍ମନ'], 'answer_start': [263]}
{'text': ['ହିନ୍ଦୀ କବି'], 'answer_start': [310]}
{'text': ['ବିପ୍ଳବ ବଞ୍ଚିରହୁ'], 'answer_start': [661]}
{'text': ['ଅମିତାଭ'], 'answer_start': [0]}
{'text': ['ଉତ୍ତର ପ୍ରଦେଶର ପ୍ରତାପଗଡ଼ ଜିଲ୍ଲାରେ ଥିବା ରାଣୀଗଞ୍ଜ ତାଲୁକାର ବାବୁପତି ଗ୍ରାମରେ'], 'answer_start': [170]}
{'text': ['ଉତ୍ତର ପ୍ରଦେଶ ରାଜ୍ୟରେ ଥିବା ଆଲ୍ଲାହାବାଦଠାରେ'], 'answer_start': [52]}
{'text': ['ପଞ୍ଜାବୀ ଶିଖ୍'], 'answer_start': [392]}
{'text': ['୧୯୪୨ ମସିହା'], 'answer_start': [13]}
```

```
[ ] for example in dataset['test']:
        sequence_values = example['context']
        print(sequence_values)
```

```
"ବକ୍ତାମାନେ ସଂଗଠନର ସଦସ୍ୟମାନଙ୍କୁ କଂଗ୍ରେସ ଆନ୍ଦୋଳନରୁ ଦୂରେଇ ରହିବାକୁ ଅନୁରୋଧ କରିଥିଲେ ଏବଂ ଏହି ନିର୍ଦ୍ଦେଶନାମା ସାଧାରଣତ observ
"ବକ୍ତାମାନେ ସଂଗଠନର ସଦସ୍ୟମାନଙ୍କୁ କଂଗ୍ରେସ ଆନ୍ଦୋଳନରୁ ଦୂରେଇ ରହିବାକୁ ଅନୁରୋଧ କରିଥିଲେ ଏବଂ ଏହି ନିର୍ଦ୍ଦେଶନାମା ସାଧାରଣତ observ
"ବକ୍ତାମାନେ ସଂଗଠନର ସଦସ୍ୟମାନଙ୍କୁ କଂଗ୍ରେସ ଆନ୍ଦୋଳନରୁ ଦୂରେଇ ରହିବାକୁ ଅନୁରୋଧ କରିଥିଲେ ଏବଂ ଏହି ନିର୍ଦ୍ଦେଶନାମା ସାଧାରଣତ observ
"ବକ୍ତାମାନେ ସଂଗଠନର ସଦସ୍ୟମାନଙ୍କୁ କଂଗ୍ରେସ ଆନ୍ଦୋଳନରୁ ଦୂରେଇ ରହିବାକୁ ଅନୁରୋଧ କରିଥିଲେ ଏବଂ ଏହି ନିର୍ଦ୍ଦେଶନାମା ସାଧାରଣତ observ
"ବକ୍ତାମାନେ ସଂଗଠନର ସଦସ୍ୟମାନଙ୍କୁ କଂଗ୍ରେସ ଆନ୍ଦୋଳନରୁ ଦୂରେଇ ରହିବାକୁ ଅନୁରୋଧ କରିଥିଲେ ଏବଂ ଏହି ନିର୍ଦ୍ଦେଶନାମା ସାଧାରଣତ observ
-ଓଡ଼ିଶାରେ ମିସନାରୀମାନଙ୍କର କାର୍ଯ୍ୟକଳାପ ଯେତେବେଳେ ସକ୍ରିୟ ହୋଇ ଉଠିଥିଲା, ସେତେବେଳେ ଆଧୁନିକଧରଣର ବିଦ୍ୟାଳୟ ପ୍ରତିଷ୍ଠା ସେମାନଙ୍କର
-ଓଡ଼ିଶାରେ ମିସନାରୀମାନଙ୍କର କାର୍ଯ୍ୟକଳାପ ଯେତେବେଳେ ସକ୍ରିୟ ହୋଇ ଉଠିଥିଲା, ସେତେବେଳେ ଆଧୁନିକଧରଣର ବିଦ୍ୟାଳୟ ପ୍ରତିଷ୍ଠା ସେମାନଙ୍କର
-ଓଡ଼ିଶାରେ ମିସନାରୀମାନଙ୍କର କାର୍ଯ୍ୟକଳାପ ଯେତେବେଳେ ସକ୍ରିୟ ହୋଇ ଉଠିଥିଲା, ସେତେବେଳେ ଆଧୁନିକଧରଣର ବିଦ୍ୟାଳୟ ପ୍ରତିଷ୍ଠା ସେମାନଙ୍କର
-ଓଡ଼ିଶାରେ ମିସନାରୀମାନଙ୍କର କାର୍ଯ୍ୟକଳାପ ଯେତେବେଳେ ସକ୍ରିୟ ହୋଇ ଉଠିଥିଲା, ସେତେବେଳେ ଆଧୁନିକଧରଣର ବିଦ୍ୟାଳୟ ପ୍ରତିଷ୍ଠା ସେମାନଙ୍କର
-ଓଡ଼ିଶାରେ ମିସନାରୀମାନଙ୍କର କାର୍ଯ୍ୟକଳାପ ଯେତେବେଳେ ସକ୍ରିୟ ହୋଇ ଉଠିଥିଲା, ସେତେବେଳେ ଆଧୁନିକଧରଣର ବିଦ୍ୟାଳୟ ପ୍ରତିଷ୍ଠା ସେମାନଙ୍କର
-ଓଡ଼ିଶାରେ ମିସନାରୀମାନଙ୍କର କାର୍ଯ୍ୟକଳାପ ଯେତେବେଳେ ସକ୍ରିୟ ହୋଇ ଉଠିଥିଲା, ସେତେବେଳେ ଆଧୁନିକଧରଣର ବିଦ୍ୟାଳୟ ପ୍ରତିଷ୍ଠା ସେମାନଙ୍କର
-ଓଡ଼ିଶାରେ ମିସନାରୀମାନଙ୍କର କାର୍ଯ୍ୟକଳାପ ଯେତେବେଳେ ସକ୍ରିୟ ହୋଇ ଉଠିଥିଲା, ସେତେବେଳେ ଆଧୁନିକଧରଣର ବିଦ୍ୟାଳୟ ପ୍ରତିଷ୍ଠା ସେମାନଙ୍କର
[୬]ଏହାଛଡା କରୋନା ସଂକ୍ରମଣ ଏବଂ ତଜ୍ଜନିତ ତାଲାବନ୍ଦ ଭଳି ଅଭାବନୀୟ ପରିସ୍ଥିତିରେ ଅତ୍ୟାବଶ୍ୟକୀୟ ସାମଗ୍ରୀର ଆମଦାନି ତଥା ରପ୍ତାନୀ ନିୟନ୍ତ୍ରଣ
[୬]ଏହାଛଡା କରୋନା ସଂକ୍ରମଣ ଏବଂ ତଜ୍ଜନିତ ତାଲାବନ୍ଦ ଭଳି ଅଭାବନୀୟ ପରିସ୍ଥିତିରେ ଅତ୍ୟାବଶ୍ୟକୀୟ ସାମଗ୍ରୀର ଆମଦାନି ତଥା ରପ୍ତାନୀ ନିୟନ୍ତ୍ରଣ
[୬]ଏହାଛଡା କରୋନା ସଂକ୍ରମଣ ଏବଂ ତଜ୍ଜନିତ ତାଲାବନ୍ଦ ଭଳି ଅଭାବନୀୟ ପରିସ୍ଥିତିରେ ଅତ୍ୟାବଶ୍ୟକୀୟ ସାମଗ୍ରୀର ଆମଦାନି ତଥା ରପ୍ତାନୀ ନିୟନ୍ତ୍ରଣ
[୬]ଏହାଛଡା କରୋନା ସଂକ୍ରମଣ ଏବଂ ତଜ୍ଜନିତ ତାଲାବନ୍ଦ ଭଳି ଅଭାବନୀୟ ପରିସ୍ଥିତିରେ ଅତ୍ୟାବଶ୍ୟକୀୟ ସାମଗ୍ରୀର ଆମଦାନି ତଥା ରପ୍ତାନୀ ନିୟନ୍ତ୍ରଣ
[୬]ଏହାଛଡା କରୋନା ସଂକ୍ରମଣ ଏବଂ ତଜ୍ଜନିତ ତାଲାବନ୍ଦ ଭଳି ଅଭାବନୀୟ ପରିସ୍ଥିତିରେ ଅତ୍ୟାବଶ୍ୟକୀୟ ସାମଗ୍ରୀର ଆମଦାନି ତଥା ରପ୍ତାନୀ ନିୟନ୍ତ୍ରଣ
ଅଂକୋର୍ ୱାଟ୍ ଏହି ଅଞ୍ଚଳର ଅନ୍ୟ ମନ୍ଦିରମାନଙ୍କଠାରୁ ଗୋଟିଏ ବିଷୟରେ ସମ୍ପୂର୍ଣ୍ଣ ଭିନ୍ନ । ତାହା ହେଲା ସମସ୍ତ ମନ୍ଦିରର ମୁଖ୍ୟଦ୍ୱାର ପୂର୍ବ ଦିଗରେ ଥିଲା
ଅଂକୋର୍ ୱାଟ୍ ଏହି ଅଞ୍ଚଳର ଅନ୍ୟ ମନ୍ଦିରମାନଙ୍କଠାରୁ ଗୋଟିଏ ବିଷୟରେ ସମ୍ପୂର୍ଣ୍ଣ ଭିନ୍ନ । ତାହା ହେଲା ସମସ୍ତ ମନ୍ଦିରର ମୁଖ୍ୟଦ୍ୱାର ପୂର୍ବ ଦିଗରେ ଥିଲା
ଅଂକୋର୍ ୱାଟ୍ ଏହି ଅଞ୍ଚଳର ଅନ୍ୟ ମନ୍ଦିରମାନଙ୍କଠାରୁ ଗୋଟିଏ ବିଷୟରେ ସମ୍ପୂର୍ଣ୍ଣ ଭିନ୍ନ । ତାହା ହେଲା ସମସ୍ତ ମନ୍ଦିରର ମୁଖ୍ୟଦ୍ୱାର ପୂର୍ବ ଦିଗରେ ଥିଲା
ଅଂକୋର୍ ୱାଟ୍ ଏହି ଅଞ୍ଚଳର ଅନ୍ୟ ମନ୍ଦିରମାନଙ୍କଠାରୁ ଗୋଟିଏ ବିଷୟରେ ସମ୍ପୂର୍ଣ୍ଣ ଭିନ୍ନ । ତାହା ହେଲା ସମସ୍ତ ମନ୍ଦିରର ମୁଖ୍ୟଦ୍ୱାର ପୂର୍ବ ଦିଗରେ ଥିଲା
ଅମିତାଭ ବଚ୍ଚନ ୧୯୪୨ ମସିହା ଅକ୍ଟୋବର ମାସ ୧୧ ତାରିଖରେ ଭାରତର ଉତ୍ତର ପ୍ରଦେଶ ରାଜ୍ୟରେ ଥିବା ଆଲ୍ଲାହାବାଦଠାରେ ଜନ୍ମ ଗ୍ରହଣ କରିଥିଲେ ।
ଅମିତାଭ ବଚ୍ଚନ ୧୯୪୨ ମସିହା ଅକ୍ଟୋବର ମାସ ୧୧ ତାରିଖରେ ଭାରତର ଉତ୍ତର ପ୍ରଦେଶ ରାଜ୍ୟରେ ଥିବା ଆଲ୍ଲାହାବାଦଠାରେ ଜନ୍ମ ଗ୍ରହଣ କରିଥିଲେ ।
ଅମିତାଭ ବଚ୍ଚନ ୧୯୪୨ ମସିହା ଅକ୍ଟୋବର ମାସ ୧୧ ତାରିଖରେ ଭାରତର ଉତ୍ତର ପ୍ରଦେଶ ରାଜ୍ୟରେ ଥିବା ଆଲ୍ଲାହାବାଦଠାରେ ଜନ୍ମ ଗ୍ରହଣ କରିଥିଲେ ।
ଅମିତାଭ ବଚ୍ଚନ ୧୯୪୨ ମସିହା ଅକ୍ଟୋବର ମାସ ୧୧ ତାରିଖରେ ଭାରତର ଉତ୍ତର ପ୍ରଦେଶ ରାଜ୍ୟରେ ଥିବା ଆଲ୍ଲାହାବାଦଠାରେ ଜନ୍ମ ଗ୍ରହଣ କରିଥିଲେ ।
ଅମିତାଭ ବଚ୍ଚନ ୧୯୪୨ ମସିହା ଅକ୍ଟୋବର ମାସ ୧୧ ତାରିଖରେ ଭାରତର ଉତ୍ତର ପ୍ରଦେଶ ରାଜ୍ୟରେ ଥିବା ଆଲ୍ଲାହାବାଦଠାରେ ଜନ୍ମ ଗ୍ରହଣ କରିଥିଲେ ।
ଅମିତାଭ ବଚ୍ଚନ ୧୯୪୨ ମସିହା ଅକ୍ଟୋବର ମାସ ୧୧ ତାରିଖରେ ଭାରତର ଉତ୍ତର ପ୍ରଦେଶ ରାଜ୍ୟରେ ଥିବା ଆଲ୍ଲାହାବାଦଠାରେ ଜନ୍ମ ଗ୍ରହଣ କରିଥିଲେ ।
ଅମିତାଭ ବଚ୍ଚନ ୧୯୪୨ ମସିହା ଅକ୍ଟୋବର ମାସ ୧୧ ତାରିଖରେ ଭାରତର ଉତ୍ତର ପ୍ରଦେଶ ରାଜ୍ୟରେ ଥିବା ଆଲ୍ଲାହାବାଦଠାରେ ଜନ୍ମ ଗ୍ରହଣ କରିଥିଲେ ।
ଅମିତାଭ ବଚ୍ଚନଙ୍କ ପିତା ୨୦୦୩ ମସିହାରେ ମୃତ୍ୟୁବରଣ କରିଥିଲେ ଓ ତାଙ୍କ ମାତା ୨୦୦୭ ମସିହାରେ ମୃତ୍ୟୁବରଣ କରିଥିଲେ । ଅମିତାଭ ବଚ୍ଚନ ନୈତି
```

TOKENIZATION OF DATA

+ Code + Text

MODEL GOOGLEFLAN-T5-BASE

```
[ ] from transformers import T5Tokenizer, T5ForConditionalGeneration

    # Load the tokenizer and model
    tokenizer = T5Tokenizer.from_pretrained("google/flan-t5-base")
    model = T5ForConditionalGeneration.from_pretrained("google/flan-t5-base", device_map="auto")

    def tokenize_function(examples):
        # Concatenate the context and question to form the input for the model
        inputs = [f"question: {q} context: {c}" for q, c in zip(examples['question'], examples['context'])]
        # Tokenize the inputs
        model_inputs = tokenizer(inputs, padding="max_length", truncation=True, max_length=128)

        # Extract the correct answer from the 'answers' field
        # Assuming the 'answers' field is a list of dictionaries with 'text' as the key for the answer text
        # and 'answer_start' as the key for the start index of the answer in the context
        # Here we are using the first answer in the list
        answers = [answer['text'][0] for answer in examples['answers']]

        # Tokenize the answers
        labels = tokenizer(text_target=answers, padding="max_length", truncation=True, max_length=128)

        # Set the labels to the tokenized answers
        model_inputs["labels"] = labels["input_ids"]
        return model_inputs

    # Assuming `dataset_dict` is your DatasetDict with the 'test' split
    tokenized_datasets = dataset.map(tokenize_function, batched=True)
```

```
tokenizer_config.json: 100% 2.54k/2.54k [00:00<00:00, 61.4kB/s]
spiece.model: 100% 792k/792k [00:00<00:00, 1.19MB/s]
special_tokens_map.json: 100% 2.20k/2.20k [00:00<00:00, 118kB/s]
tokenizer.json: 100% 2.42M/2.42M [00:00<00:00, 32.0MB/s]
You are using the default legacy behaviour of the <class 'transformers.models.t5.tokenization_t5.T5Tokenizer'>. This
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-tuned or trained.
config.json: 100% 1.40k/1.40k [00:00<00:00, 80.0kB/s]
model.safetensors: 100% 990M/990M [00:26<00:00, 36.5MB/s]
generation_config.json: 100% 147/147 [00:00<00:00, 5.23kB/s]
Map: 100% 1680/1680 [00:02<00:00, 698.56 examples/s]
```

```
[ ] tokenized_datasets

DatasetDict({
    test: Dataset({
        features: ['id', 'context', 'question', 'answers', 'input_ids', 'attention_mask', 'labels'],
        num_rows: 1680
    })
})
```

```
[ ] from transformers import Seq2SeqTrainingArguments, Seq2SeqTrainer, DataCollatorForSeq2Seq
    import torch
    from datasets import load_metric, load_dataset
    from peft import get_peft_model, LoraConfig
```

```
No CUDA runtime is found, using CUDA_HOME='/usr/local/cuda'
```

TRAINING THE MODEL BY SEQ2SEQTRAINING

```
lora_config = LoraConfig(
r = 4,
lora_alpha = 32,
lora_dropout = 0.1,
bias = "none"
)

model = get_peft_model(model, lora_config)

training_args = Seq2SeqTrainingArguments(
    output_dir="/kaggle/working",
    evaluation_strategy="steps",
    learning_rate=2e-5,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    weight_decay=0.01,
    save_total_limit=3,
    num_train_epochs=3,
    predict_with_generate=True,
    fp16=torch.cuda.is_available(),
    report_to="none",
    save_steps=500,
    logging_steps=100,
    load_best_model_at_end=True,
)

# Prepare data collator
data_collator = DataCollatorForSeq2Seq(tokenizer, model=model)

# Prepare metric for evaluation with trust_remote_code=True
metric = load_metric("sacrebleu", trust_remote_code=True)
```

```python
def compute_metrics(eval_pred):
    predictions, labels = eval_pred
    decoded_preds = tokenizer.batch_decode(predictions, skip_special_tokens=True)
    decoded_labels = tokenizer.batch_decode(labels, skip_special_tokens=True)

    decoded_preds = [pred.strip() for pred in decoded_preds]
    decoded_labels = [[label.strip()] for label in decoded_labels]

    result = metric.compute(predictions=decoded_preds, references=decoded_labels)
    result = {"bleu": result["score"]}
    return result
```

```
/usr/local/lib/python3.10/dist-packages/transformers/training_args.py:1474: FutureWarning: `evaluation_strategy` is
  warnings.warn(
<ipython-input-11-61d826630f05>:33: FutureWarning: load_metric is deprecated and will be removed in the next major v
  metric = load_metric("sacrebleu", trust_remote_code=True)
Downloading builder script: 7.65k/? [00:00<00:00, 360kB/s]
```

```
[ ] tokenized_datasets
```

```
DatasetDict({
    test: Dataset({
        features: ['id', 'context', 'question', 'answers', 'input_ids', 'attention_mask', 'labels'],
        num_rows: 1680
    })
})
```

```
[ ] tokenized_datasets['test']
```

```
Dataset({
    features: ['id', 'context', 'question', 'answers', 'input_ids', 'attention_mask', 'labels'],
    num_rows: 1680
})
```

```
[ ] from sklearn.model_selection import train_test_split

    # Split the 'test' dataset into train and validation sets
    split_datasets = tokenized_datasets['test'].train_test_split(test_size=0.2)
    train_dataset = split_datasets['train']
    val_dataset = split_datasets['test']  # This is the validation set
```

```
[ ] len(train_dataset)
```

```
1344
```

```
[ ] len(val_dataset)
```

```
336
```

```
[ ] train_dataset[5:10]
```

```
{'id': ['227', '1515', '267', '319', '725'],
 'context': ['କରୋନା ଭୁଟାଣୁ ସଂକ୍ରମଣ ସମୟରେ ଗରିବ ଲୋକଙ୍କ ଜୀବନଜୀବିକା ବିଶେଷ ଭାବେ ପ୍ରଭାବିତ ହେଉଥିବାରୁ ନବୀନ ପଟ୍ଟନାୟକ ୨୨୦୦ କୋଟି
ଟଙ୍କାର ଏକ ପ୍ୟାକେଜ୍ ଘୋଷଣା କରିଥିଲେ । ଏଥିରେ ନିର୍ମାଣ ଶ୍ରମିକଙ୍କ ସମେତ ଖାଦ୍ୟ ସୁରକ୍ଷା ଯୋଜନା ଓ ସାମାଜିକ କଲ୍ୟାଣ ଯୋଜନାର ହିତାଧିକାରୀଙ୍କୁ ଆର୍ଥିକ
ସହାୟତା ପ୍ରଦାନ କରାଯିବ । ତାଲାବନ୍ଦ ଯୋଗୁଁ ଅନ୍ୟ ରାଜ୍ୟରେ ଅଟକି ରହିଥିବା ଓଡ଼ିଆମାନଙ୍କୁ ଖାଦ୍ୟ ଓ ଔଷଧ ଯୋଗାଇ ଦେବାକୁ ଓଡ଼ିଶା ସରକାର ଅନ୍ୟ ରାଜ୍ୟର
ସ୍ୱେଚ୍ଛାସେବୀ ଅନୁଷ୍ଠାନଗୁଡିକୁ ଅନୁରୋଧ କରିଛନ୍ତି । ଏହି ଖର୍ଚ୍ଚ ଓଡ଼ିଶା ସରକାର ଭରଣା କରିବେ । ଓଡ଼ିଶା ଉଚ୍ଚ ନ୍ୟାୟାଳୟ ପ୍ରବାସରୁ ଫେରୁଥିବା ଓଡ଼ିଆମାନଙ୍କୁ ସାମାନ୍ତରେ
ଅଟକାଇବା ସହ ସେଠାରେ ସେମାନଙ୍କ ରହିବା ଓ ଖାଇବା ବ୍ୟବସ୍ଥା କରିବାକୁ କରିବାକୁ ନିର୍ଦ୍ଦେଶ ଦେଇଥିଲେ । [୧୩]ଚଳିତ ଖରିଫ୍ ଓ ଆସନ୍ତା ରବି ଫସଲ ପାଇଁ
ରାଜ୍ୟରେ କୃଷି କାର୍ଯ୍ୟ ଜାରି ରହିବା ସହ କୃଷିକାର୍ଯ୍ୟ ପାଇଁ ବିଭିନ୍ନ କୃଷି ସାମଗ୍ରୀ, ସାର, କୀଟନାଶକ ଔଷଧ ଓ ଯନ୍ତ୍ରପାତି ଖ୍ଲାନାନ୍ତରଣରେ କୌଣସି ବାଧା ରହିବ ନାହିଁ
ବୋଲି ସରକାର ସୂଚନା ଦେଇଛନ୍ତି । [୧୪] ଲକ୍ ଡାଉନ ଓ ସଟ ଡାଉନଜନାଳାବନ୍ଦ ସମୟରେ ବିଭିନ୍ନ ଅତ୍ୟାବଶ୍ୟକୀୟ ସେବା ଯଥା ବିଜୁଳି, ଜଳଯୋଗାଣ, ହାସପାତାଳ,
ଔଷଧ ଦୋକାନ, ଅଗ୍ନିଶମ ସେବା ଇତ୍ୟାଦି ଜାରି ରହିବା ସମେତ ଜାତୀୟ ରାଜପଥ ଓ ରାଜ୍ୟ ରାଜପଥ କଡ଼ରେ ଥିବା ଢାବା ଓ ଗ୍ୟାରେଜ ଖୋଲା ରହିବ ଓ ସରକାରୀ
ସେସରକାରୀ କାର୍ଯ୍ୟାଳୟ, ଶିଳ୍ପ, କଳ କାରଖାନା,ଶିକ୍ଷାନୁଷ୍ଠାନ, ଧାର୍ମିକ ଅନୁଷ୍ଠାନ, ଅଦାଲତ, ବାଣିଜ୍ୟିକ ଅନୁଷ୍ଠା [୧୫]ରାଜ୍ୟ ସରକାର ଲକଡାଉନ ଅବଧିରୁ ଏପ୍ରିଲ ୩୦
ଯାଏଁ ବୃଦ୍ଧି କରିବା ସହ କୃଷି ତଥା ଅନ୍ୟାନ୍ୟ ଆନୁଷଙ୍ଗିକ କ୍ଷେତ୍ରରେ ଏହି କଟକଣାକୁ କିଛି ମାତ୍ରାରେ କୋହଳ କରିଛନ୍ତି ।',
 'ଯୁଜୀନ ଏମ. ମକରଙ୍କ ଅନୁସାରେ , ଭାରତୀୟ ପାରମ୍ପରିକ ସଂସ୍କୃତି ଅପେକ୍ଷାକୃତ କଠୋର ସାମାଜିକ ପଦନ୍ତୁକ୍ରମାବାରା ପରିଭାଷିତ କରାଯାଇଛି । ସେ ଏହା ମଧ୍ୟ
କହିଥିଲେ ଯେ , ପିଲାମାନଙ୍କୁ ଛୋଟ ବୟସରେ ହିଁ ସେମାନଙ୍କ ଭୂମିକା ଏବଂ ସମାଜରେ ଚାଙ୍କ ସ୍ଥାନ ସମ୍ବନ୍ଧରେ କୁହାଯାଇଥାଏ । ଚାଙ୍କର ଏହି କଥାରେ ଆହୁରି ମଧ୍ୟ ଆଜ
ମିଳିଥାଏ ଏବଂ ତାହାର ସ୍ୱରୂପ ଏଡିଯେ , ବହୁତ ଲୋକ ଏହି କଥାକୁ ମାନନ୍ତି ଯେ ଚାଙ୍କର ଜୀବନକୁ ନିର୍ଦ୍ଧାରଣ କରିବାରେ ଦେବତା ଏବଂ ଆତ୍ମା ମାନଙ୍କର ହିଁ ପୂର୍ଣ୍ଣ ଭୂମିକା
ହୋଇଥାଏ । ଧର୍ମ ବିଭାଜିତ ସଂସ୍କୃତି ଯେପରି ବହୁ ମତଭେଦ ; ଯେମିତିକି , ସେମାନଙ୍କ ଠାରୁ ବହୁ ଅଧିକ ଶକ୍ତିଶାଳୀ ବିଭାଜନ ହୋଇଥାଏ ହିନ୍ଦୁ ପରମ୍ପରାରେ , ମାନ୍ୟତ
ଅପ୍ରଦୂଷିତ ଏବଂ ପ୍ରଦୂଷିତ ବ୍ୟବସାୟର । ସଖ୍ଯ ସାମାଜିକ ଅମାନ୍ୟ ଲୋକ ଏହି ହଜାରେ ଲୋକଙ୍କର ସମୂହକୁ ନିୟନ୍ତ୍ରିତ କରିଥାନ୍ତି । ସହରରେ ଏଥିମଧ୍ୟରୁ କିଛି ଶ୍ରେଣୀ ଧ୍ୱସ
ପରିମାଣକୁନି ଏବଂ ଆଉ କିଛି ଭଲାଇ ହୋଇସାରିଛନ୍ତି । ପାରିବାରିକ ପରିବାର (Nuclear family) ଭାରତୀୟ ସଂସ୍କୃତି ପାଇଁ ନୂତନ ଅଟେ । ମହତ୍ୱପୂର୍ଣ୍ଣ ପାରିବାରିକ
```

FINE TUNING

```
[ ] trainer = Seq2SeqTrainer(
        model=model,
        args=training_args,
        train_dataset=train_dataset,
        eval_dataset=val_dataset,
        tokenizer=tokenizer,
        data_collator=data_collator,
        compute_metrics=compute_metrics
    )

    # Fine-tune the model
    trainer.train()
```

[504/504 04:44, Epoch 3/3]

Step	Training Loss	Validation Loss	Bleu
100	0.000000	nan	0.000000
200	0.000000	nan	0.000000
300	0.000000	nan	0.000000
400	0.000000	nan	0.000000
500	0.000000	nan	0.000000

```
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1168: UserWarning: Using the model-agnostic
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/huggingface_hub/file_download.py:1132: FutureWarning: `resume_download` is d
  warnings.warn(
TrainOutput(global_step=504, training_loss=0.0, metrics={'train_runtime': 285.7963, 'train_samples_per_second':
14.108, 'train_steps_per_second': 1.763, 'total_flos': 691605237399552.0, 'train_loss': 0.0, 'epoch': 3.0})
```

```
[ ] # Generate predictions using the trained model
    predictions = trainer.predict(tokenized_datasets['test'])

    # Decode the predictions
    decoded_preds = tokenizer.batch_decode(predictions.predictions, skip_special_tokens=True)
```

```
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1168: UserWarning: Using the model-agnostic
  warnings.warn(
```

ACCURACY AND F1 SCORE

```
[ ] from utils import compute_score
    from utils import f1_score
    # Calculate the BLEU score
    bleu_score = compute_score(predictions=decoded_preds, references=references)
    f1 = f1_score(decoded_preds, references)
    print(f"BLEU SCORE: {bleu_score}")
    print(f"F1 score of the decoded preds: {f1}")
```

```
BLEU SCORE: 0.4050021510445334
F1 score of the decoded preds: 0.5322555704901326
```

PREDICTIONS

```
[ ]  # Print some examples
     for i, pred in enumerate(decoded_preds[:5]):
         print(f"Example {i + 1}:")
         print(f"Prediction: {tokenized_datasets['test'][i]['question']}")
         print(f"Reference: {tokenized_datasets['test'][i]['answers']}")
```

```
Example 1:
Prediction: କେଉଁ ସରକାର ରିପୋର୍ଟ କରିଛନ୍ତି ଯେ ଆରଏସଏସ କୌଣସି ସରକାରୀ ଆଦେଶକୁ ଉଲ୍ଲଂଘନ କରି ନାହାନ୍ତି?
Reference: {'text': ['ବ୍ରିଟିଶ ସରକାର'], 'answer_start': [717]}
Example 2:
Prediction: କେଉଁ ବର୍ଷରେ ଆରଏସଏସ ନେତାମାନେ ସମସ୍ତ କାର୍ଯ୍ୟକଳାପରୁ ଦୂରେଇ ରହିଥିଲେ?
Reference: {'text': ['୧୯୪୨ ମସିହାରେ'], 'answer_start': [413]}
Example 3:
Prediction: ଆରଏସଏସର କେଉଁ ନେତା କହିଥିଲେ ଯେ ଆରଏସଏସ ଭାରତ ଛାଡିବା ଆନ୍ଦୋଳନକୁ ସମର୍ଥନ କରୁନାହିଁ?
Reference: {'text': ['ଏମ.ଏସ. ଗୋଲୱାଲକର'], 'answer_start': [169]}
Example 4:
Prediction: କାର୍ଯ୍ୟକଳାପରୁ ନିବୃତ୍ତ ହେବାକୁ ଆରଏସଏସ ନେତାମାନଙ୍କୁ କିଏ ନିର୍ଦ୍ଦେଶ ଦେଇଛନ୍ତି?
Reference: {'text': ['ଏମ.ଏସ. ଗୋଲୱାଲକର'], 'answer_start': [169]}
Example 5:
Prediction: ବ୍ରିଟିଶ ସରକାର ଦର୍ଶାଇଛନ୍ତି ଯେ ଆରଏସଏସ କେଉଁମାନଙ୍କ ବିରୁଦ୍ଧରେ କୌଣସି ନାଗରିକ ଅବମାନନାକୁ ସମର୍ଥନ କରୁନାହାନ୍ତି?
Reference: {'text': ['ବ୍ରିଟିଶ ସରକାର'], 'answer_start': [718]}
```

MODEL SAVED

```
[ ]  model.save_pretrained('./models')
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/file_download.py:1132: FutureWarning: `resume_download` is de
  warnings.warn(
```