Technical Documentation
Question Answering System Based on BERT

Contents

1 Introduction

This document describes the technologies used in an extractive *Question Answering* (QA) project, as well as the general structure of the project and how it can be executed. The goal of the project is to develop a system that can answer questions by extracting a relevant fragment from a given text, using modern natural language processing techniques.

The main technologies used are:

- BERT and the Hugging Face Transformers library
- spaCy
- NLTK
- scikit-learn
- Hugging Face Datasets and Evaluate

2 BERT (Bidirectional Encoder Representations from Transformers)

BERT is a language model based on the Transformer architecture, developed by Google. Its main characteristics are:

- It is a bidirectional model, meaning it analyzes context from both the left and the right of a word simultaneously.
- It is pre-trained on very large amounts of text using tasks such as:
  - Masked Language Modeling (MLM)
  - Next Sentence Prediction (NSP)
- It can be fine-tuned for specific tasks such as:
  - Text classification
  - Named Entity Recognition
  - Question Answering

In this project, BERT is used for the extractive *Question Answering* task, where the model predicts the start and end positions of the answer within a text.

3 Hugging Face Transformers

The Hugging Face Transformers library provides:

- Standardized implementations for Transformer models (BERT, RoBERTa, DistilBERT, GPT, etc.)
- Optimized tokenizers
- Classes for training and evaluation
- Ready-to-use pipelines for inference

Advantages:

- Allows fast loading of pre-trained models
- Simplifies the fine-tuning process
- Offers a large ecosystem of models and datasets

In the project, Hugging Face is used for:

- Loading the BERT model
- Tokenizing text
- Training and evaluation
- Running inference through the QA pipeline

4 spaCy

spaCy is a natural language processing library focused on real-world applications.

Main features:

- Tokenization
- Sentence segmentation
- Lemmatization
- Syntactic parsing
- Named Entity Recognition (NER)

Characteristics:

- Very fast and optimized

- Provides pre-trained models for multiple languages
- Easy to integrate into applications

In the project, spaCy is used for:

- Cleaning and normalizing text
- Removing invalid characters
- Light processing without altering text positions (important for extractive QA)

5 NLTK (Natural Language Toolkit)

NLTK is one of the oldest and most well-known NLP libraries in Python.

Features:

- Tokenization
- Sentence segmentation
- Stemming and lemmatization
- Stopwords
- Linguistic resources

Advantages:

- Very good for educational purposes
- Offers many classical NLP algorithms

In the project, NLTK is used for:

- Segmenting context into sentences
- Analysis and debugging
- Possible splitting of long texts into chunks

6 scikit-learn

scikit-learn is a classical machine learning library for Python.

It provides:

- Classification, regression, and clustering algorithms
- Evaluation tools
- Data preprocessing utilities

In the project, scikit-learn is used optionally for:

- Computing additional metrics
- Statistical analysis of results
- Performance reports

7 Hugging Face Datasets

Hugging Face Datasets is a library for efficient handling of large datasets.

Features:

- Fast loading of standard datasets (SQuAD, GLUE, etc.)
- Support for map, filter, shuffle operations
- Integration with Transformers

In the project it is used for:

- Loading the SQuAD dataset
- Preprocessing and tokenizing data
- Splitting into training and validation sets

8 Hugging Face Evaluate

Evaluate is a library for computing standard metrics.

Features:

- Implementations for NLP metrics
- Support for SQuAD, BLEU, ROUGE, etc.

In the project it is used for:

- Computing Exact Match and F1 metrics for QA

9 Project Structure

The project follows a standard pipeline:

1. Data Loading
2. Preprocessing
3. Model Implementation
4. Evaluation

The logical structure is:

- Loading data (e.g., SQuAD or local data)
- Cleaning text with spaCy and NLTK
- Tokenization with Hugging Face Tokenizer
- Training the BERT model
- Evaluation using standard metrics
- Inference on new data

The main file contains:

- Preprocessing functions
- Model configuration
- Training and evaluation
- Inference for the user

10 How to Run

There are two main usage modes:

10.1 Training and Evaluation

The project is run in training mode to adapt the model to the SQuAD dataset:

- Data is loaded
- Preprocessing is applied
- The model is trained
- Metrics are computed
- The resulting model is saved

10.2 Inference

A pre-trained model is used to answer questions:

- The user provides a text and a question
- The text is cleaned
- The model predicts the answer position
- The extracted fragment is displayed

11 Conclusion

The project demonstrates the integrated use of the most popular modern NLP technologies:

- Transformer models (BERT)
- The Hugging Face ecosystem
- Classical NLP libraries (spaCy, NLTK)
- Evaluation tools (Evaluate, scikit-learn)

This combination enables rapid development of a robust Question Answering system that is easy to extend and adapt for other natural language processing tasks.