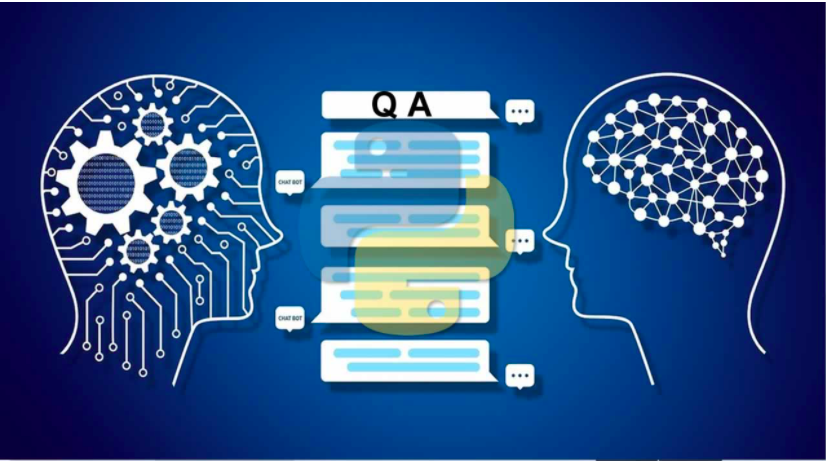
BERT- IMPLEMENTATION

Q&A SYSTEM FINAL REPORT

Ariga Tejasimha



Prepared for:

Vahid Behzadan, Ph.D.

# Table of contents

1. Problem statement
2. Approach to solve
3. BERT
4. Bugs occurred during implementation
5. Solving the bugs
6. Data processing
7. Answer retrieving
8. Future work
9. References

## 

# Problem statement

We have planned to build system where we can ask a question and get the answer relevant to the question.we will asking our query in the text format and we will get the answer in the same format.

Here we have decided to build a Question and Answering system.we have to answer couple question before building the system

1. What kind of Q&A system we are planning to build
2. What is the area of questions or data sets we need
3. We kind of model or technique we are gonna use to build the system.

Question answering system must be able to understand the context of the question and able to answer appropriately.

# Approach to solve

First we have to decide the area of questions to be asked.Let’s take wikipedia Q&A system.

In this system we can ask any question regarding any city or historic monuments in a particular place . It will answer searching from the relevant article loaded in the local wikipedia article data base we created during the Q&A system. As we are build a term project we are just storing this in the local machines. We will getting the text format of data from wiki data and store in folder name text of each city data in text format in different files. Now let’s dive into code structure and work flow of the Q&A model.

By looking the problem we can solve the question in four stages

* Stage 1:- Extraction of text from wikipedia
* Stage 2:- processing the question
* Stage 3:- Retrieving the context from the text
* Stage 4:- Retrieving the answer from the context

## Extraction of text from wikipedia

In first step we are gonna download the text or articles from the wikipedia in our local system rather than searching in the wikipedia every time. It is best thing to be done rather than searching from wikipedia every time. So we will cache the text from the wikipedia in the form of text files in a folder.we will name the each data file or article in the same of city related. We are getting the data in the text format itself which will be easy to manipulate.

## Processing the Question

After we are done with getting the data and storing it in a folder. We are going into most important part of project “**Processing**”. In this stage we will get the important pieces from the question we have asked to the system.we will extract the important query to look up in the dataset of articles or text files.

## Retrieving the context

The input or important question , we will try to find the most relevant sentences from the data corpus we have created in local machine.This will help have a small search space for our answers and will lead to a higher accuracy of correct answering system.

## Retrieving the answer from context

After creating the search space we have to just get the correct answer now. This where we will be using the BERT. We will be discussing about it in the future discussion. We will be using a smaller , light weight version of BERT which is called Distilled Bert to get the best answer and relevant ones. We will feed the context from the earlier step to our model and will get our answer in return.

# BERT

Let’s discuss about BERT. **Bidirectional Encoder Representations from Transformers** (**BERT**) is a Transformer based machine learning technique for natural language processing pre-training developed by google.The original English-language BERT has two models:

1. The BERTBASE: 12 Encoders with 12 bidirectional self-attention heads,
2. The BERTLARGE: 24 Encoders with 24 bidirectional self-attention heads. The Both models are pre-trained from unlabeled data extracted from the Book Corpus with 800M words and English wikipedia with 2,500M words.

On October 25, 2019, Google Search announced that they had started applying BERT models for English Language search queries within the US. On December 9, 2019, it was reported that BERT had been adopted by Google Search for over 70 languages. In October 2020, almost every single English-based query was processed by BERT.

But we will be using DistilBERT instead of original because DistilBERT is a simpler, more lightweight and faster version of Google's BERT model and it was developed by HuggingFace. It runs faster than the original model because it has much less parameters but it still keeps most of the original model performance.

# Bugs occurred during implementation

Though we want to build a question answering system but it was difficult for us to build it on wiki data. There was no prior woke done on it. So we have been building it from scratch. We have gone some youtube videos how BERT has been used on SQuAD(Stanford Question Answering Dataset).

But there were some implementation bugs we faced.

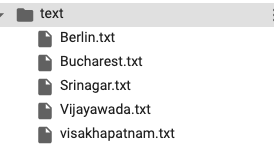
1. Installing of em\_core\_web\_sm
2. Gensim versioning error
3. Installation and versioning error in Transformers

# Solving of bugs

Though it was a challenging task to find the perfect versions and installation process to use them without error.

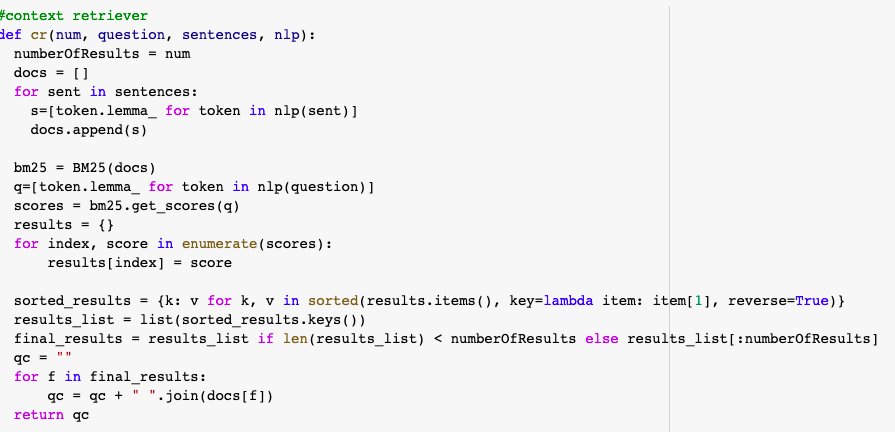
For torch we have use 1.4.0 and transformer == 2.8.0 which we are gonna use for the question answering system.

# Data Processing

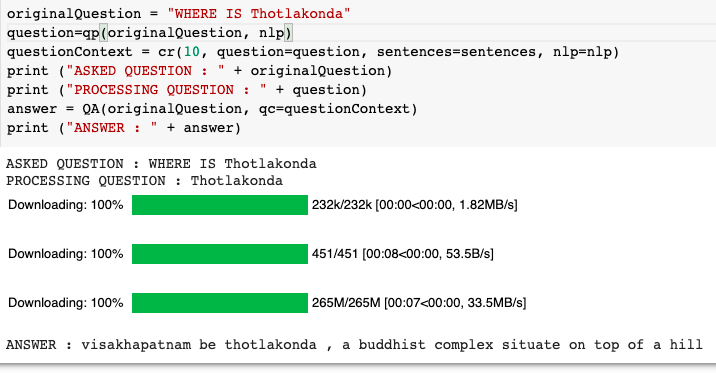
For acquiring of data what we have done is we have used wikipedia .so we have to goto Wikidata page and the city name and it’s article number for example for Visakhapatnam it is Q200016. We can any wikipedia article related data by going to wiki data page search the city you will get the text data of the wikipedia. which we are interested in. As we have discussed before data is stored in folder. Now our biggest task will be getting the relevant answer from the files. We will be discussing the getting the answer retrieval.

Here we got the data from wikipedia and create a kind of data base where we can go and search. While we got processing question we have to go for context retrieval where we use the **BM25**

BM25 :-

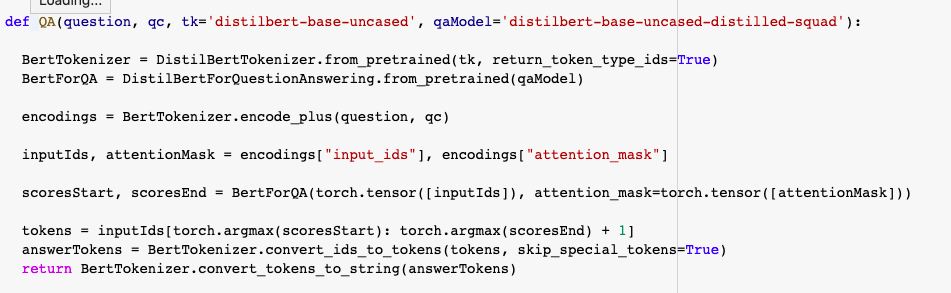
**BM25**is a function or an algorithm used to rank a list of documents based on a given query. That's why it is also called a ranking function. It is very similar to **TF-IDF** and it is actually so good that I understand it is used in ElasticSearch for document ranking.

* It is a bag-of-words model, and that means the algorithm disregards grammar structure but takes into account term frequencies - making it just ideal for our project.
* It takes a query and helps us sort a collection of documents based on how relevant they are for that query.
* The Gensim package has a very good BM25 implementation that is very easy to use.

Simple once a question is asked it goes to process and then get the processed question and then go to context r4etrival to get the context relevant to the question and then answer retrieval will get the answer for the question asked. This is the proper data flow of our question answer system.

# Answer Retrieval

In this section we will discuss about the getting answer for the processed question.



We have been using Distilbert in the section where we get the answer.

# FUTURE WORK

We can create update some features to the system. Like

* As we can add the feature like input and output I.e,we can add voice as intake and system give the answer in voice over
* We can also create a cloud and store the data get the answer from the cloud if not found then we can get the answer from google search if needed
* We can change the area of asking questions like stocks, health care and real estate. It has very wide range of applications
* We can create our own Siri, google and cortana personal mobile assistants if needed.

# References

# 

For any general queries

[www.google.com](http://www.google.com)

For BM25

<https://en.wikipedia.org/wiki/Okapi_BM25>

For Bert

<https://en.wikipedia.org/wiki/BERT_(language_model)>

For wikidata

<https://www.wikidata.org/wiki/Wikidata:Main_Page>

We have done most of our project in the colab. Which is the online jupytr complier developed by the google.

[https://colab.research.google.com/notebooks/intro.ipynb#recent=true](https://colab.research.google.com/notebooks/intro.ipynb%23recent=true)