14th Biennial International Conference and Exposition
SPG 2023

# Building a Closed Domain Question Answering System by using Natural Language Processing

*Sanjay Chakravorty*, ONGC, India*
chakravorty_sanjay@ongc.co.in

**Keywords**

Natural Language Processing, Closed Domain, Question Answering System, BERT

**Abstract**

A proof of concept Artificial Intelligence (AI) based application has been developed for a Question Answering system based on documents from Geological and Geophysical (G & G) domain by using Natural Language Processing (NLP) techniques. The present paper discusses the complete workflow of the Question Answering system, starting from data annotation, using pre-trained BERT model, training/fine tuning the BERT model with new annotated dataset, preparing the data corpus and finally developing a web based user interface to access the core application.

## Introduction

Every organisation generates a lot of information. The information can be of many types. Some of this information gets generated during operational activities. Some others are generated during research works. There will also be cumulative information belonging to a certain time interval. All the information is documented and published as reports, periodicals etc. Over a period of time organizations keeps accumulating documents creating a large repository. This repository is the real treasure of that organisation.

People working in the organisation keep referring to these documents from time to time. They refer the documents to understand about previous work done on similar subject or in same area etc. For searching information from a document one has either to read the document completely or use keyword search. If the documents to be searched are more than one there exists text based search application like 'Solr' which performs admirably. But sometimes situations may come where one may be looking for direct answer of their queries from a document repository. The solution to the above problem requires development of a question answering platform. Previously it was difficult to develop such a platform for a document repository with traditional software development approach. But the evolution of Natural Language Processing (NLP) along with Artificial Intelligence (AI) has opened up new vistas in developing software platform that attempts to address complex linguistic challenges.

The present effort is an attempt to create a Question Answering application which answers queries referring to a Geological and Geophysical (G & G) document repository belonging to oil and gas domain.

## Natural Language Processing

Natural Language Processing (NLP) refers to the branch of computer science and more specifically, the branch of artificial intelligence or AI, concerned with giving computers the ability to understand text and spoken words in much the same way as human beings can. The issues that can be addressed with NLP are as follows; classifying whole sentences, classifying each word in a sentence, generating text content, extracting an answer from a text, machine translation etc.

## Open Domain vs. Closed Domain

Open-domain systems deal with questions about almost on any topic and it relies on general ontologies and world knowledge. In closed domains of knowledge, a question of interest is not generally available for the internet to answer. Examples of closed domain are medicine, oil and gas, mining, automotive etc.

## BERT

BERT stands for Bidirectional Encoder Representations from Transformers. BERT is developed by Google. It is a language model designed for contextual understanding of words within a sentence. It allows more accurate answering of questions. For contextual understanding it uses Transformers. Transformers are an attention mechanism that learns contextual relations between words or sub-words in a sentence. Transformer includes two separate mechanisms - an encoder that reads the text input and a decoder that produces a prediction for the task. Since BERT's aim is to generate a language model, only the encoder mechanism is necessary.

In the present case BERT from HuggingFace transformers library has been used.

**Fine Tuning BERT**

BERT can be used for addressing different types of language tasks, including Question Answering by only adding a small layer to the base model. The application receives a question about a paragraph and the answer is marked in the paragraph. The model can then be trained by learning two extra vectors that mark the beginning and the end of the answer.

**System Architecture**

The present system uses cdQA-suite developed by André Macedo Farias and others (https://github.com/cdqa-suite) as its core component. Its architecture is based on two main components, the Retriever and the Reader.

When a query is raised, the Retriever is pointed to a document repository as selected by the user. The Retriever creates TF-IDF (Term Frequency - Inverse Document Frequency) features based on unigrams and bigrams and compute the cosine similarity between the question sentence and each document of the repository.

After selecting the most likely documents, the system divides each document into paragraphs and sends them with the question to the Reader. The Reader is a pre-trained Deep Learning model. The model used is the Pytorch version of BERT, which was made available by HuggingFace.

Next, the Reader outputs the most probable answer it can find in each paragraph. After the Reader, there is a final layer in the system that compares the answers by using an internal score function and outputs the most likely one according to the scores.

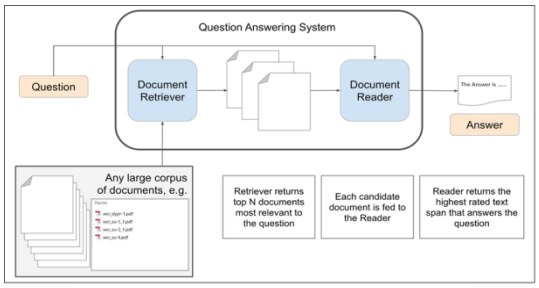



Figure 1: Question Answering system architecture

Initially the pre trained model was used as Reader. The pre trained model was trained on SQuAD (Stanford Question Answering Dataset) 1.1 dataset. Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowd workers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable.

Subsequently annotated dataset in SQuAD format was prepared from ONGC internal documents repository pertaining to Geological and Geophysical domain. A new dataset was created by merging the SQuAD 1.1 dataset and the annotated dataset. This new dataset was then used to train and fine tune the Reader.

**Data Annotation and Model Training**

As the SQuAD 1.1 dataset is based on open domain documents, it was felt the dataset can be augmented by adding annotated data from Geological and Geophysical domain documents. At first a dataset whose format was same as SQuAD format was created. Then that dataset was opened in the data annotator application. In that application data annotation was done, first by adding a question and then marking the relevant answer in the paragraph from where the question was asked. The new annotated dataset thus created was subsequently merged with existing SQuAD 1.1 dataset.

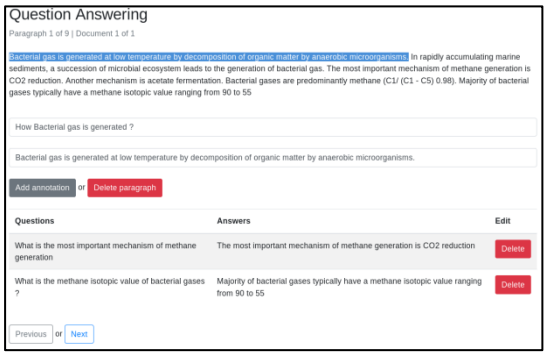



Figure 2: Data annotation

The model was then trained with the dataset for 100 epochs in a server having A100 (40 GB) GPU.

**Data Corpus preparation**

Data corpus preparation involved converting and merging PDF documents to a single data file of JSON (Java Script Object Notation) format. Two data corpuses were created. The first corpus was generated from a base of 938 documents whose cumulative size was 19 GB. The second corpus was generated from a base of 3651 documents whose cumulative size was 83 GB.

**Prediction**

The output of the question answering application constitutes of following four components

| Sl. No. | Component |
|---|---|
| 1. | Predicted Answer |
| 2. | Name of the document in which answer was found |
| 3. | The paragraph of the above document in which the answer was found |
| 4. | Internal score function used for preparing the ranking of predicted answer |

Table 2: Prediction components

The application displays those predicted answers which are having positive internal score. The answers are also ranked according to that score. Answers with higher score are more contextually relevant to the question, then those with lower score. Answers with score nearer to zero are almost irrelevant.

**Application setup and User Interface**

The application has been developed in a Linux based container environment using Python of version 3.7. A web based user interface has also been developed and connected with the backend application. In the web based application the user has to first select the relevant data corpus and then key in the query for the application to fetch the answer.

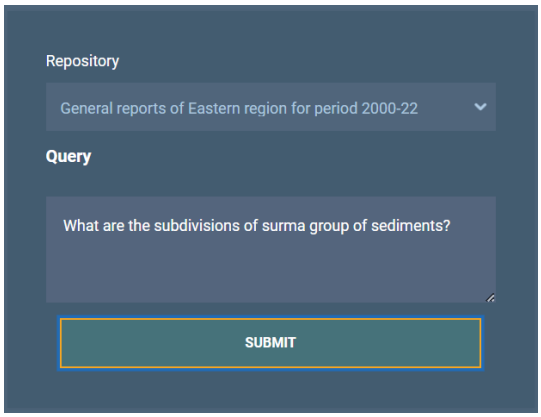



Figure 3: Data corpus selection and input question

The following figure displays the predicted answers.

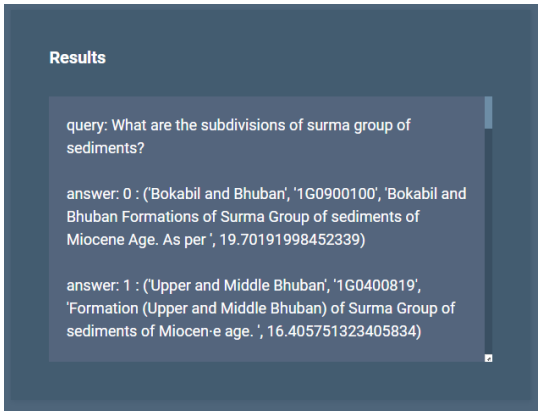



Figure 4: Predicted answer with other associated information

The next figure contains a different query with the predicted answer in the subsequent figure.

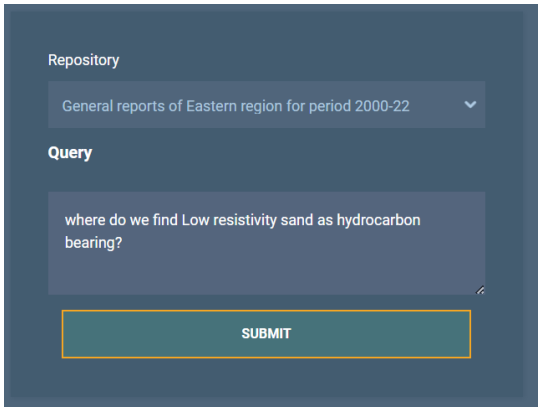



Figure 5: A different question

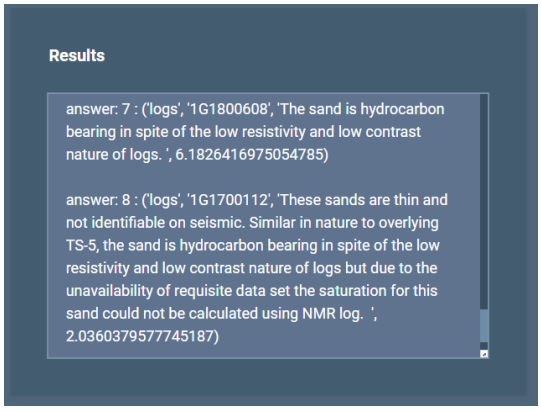
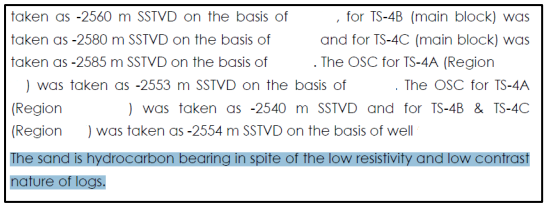



Figure 6: Predicted answer

In the following figure the highlighted text is the exact paragraph of the document from where the

Question Answering system has picked up the answer.


taken as -2560 m SSTVD on the basis of , for TS-4B (main block) was taken as -2580 m SSTVD on the basis of and for TS-4C (main block) was taken as -2585 m SSTVD on the basis of . The OSC for TS-4A (Region ) was taken as -2553 m SSTVD on the basis of . The OSC for TS-4A (Region ) was taken as -2540 m SSTVD and for TS-4B & TS-4C (Region ) was taken as -2554 m SSTVD on the basis of well

The sand is hydrocarbon bearing in spite of the low resistivity and low contrast nature of logs.

Figure 7: Exact paragraph from where the answer is picked up

## Conclusion

The application gave improved results compared to traditional way of searching information using keywords. In keyword search method, searching with multiple words or more than one group of words will result in finding documents having both words/group of words, but they may not be contextually connected. Both the group of words may be present in the same document in two different far of paragraphs and hence having no contextual correlation between them. But the AI based Question Answering system tries to give outputs which are contextually connected and hence more helpful to user's information search aim.

## References

1. Rani Horev 2018, BERT explained: State of the art language model for NLP, https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp

2. An End-To-End Closed Domain Question Answering System, 2019, https://github.com/cdqa-suite

3. André Macedo Farias 2019, How to create your own Question-Answering system easily with python, https://towardsdatascience.com/how-to-create-your-own-question-answering-system-easily-with-python

4. https://huggingface.co/docs/transformers

## Acknowledgment

The author is thankful to ONGC Management for allowing publishing of the paper in SPG-Kochi 2023 conference. The author is also thankful to Shri A.V. Satyanarayana GGM-National Head Database, ONGC, for his kind consent in using necessary facilities to carry out this work. The author expresses sincere thanks to Shri Panyam Prabhakar, CGM-Head EPINET ONGC for his constant support and valuable suggestions. The views expressed in this paper are those of author only and may not necessarily represent the formal opinion of ONGC.