Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 198 (2022) 379–384

Procedia Computer Science

www.elsevier.com/locate/procedia

The 2nd International Workshop on Artificial Intelligence for Natural Language Processing

(IA&NLP 2021)
November 1-4, 2021, Leuven, Belgium

# Bert for Question Answering applied on Covid-19

Awane Widad* , Ben Lahmar El Habib , El Falaki Ayoub

*Faculty of Sciences Ben M'sik Hassan II University, B.P 7955, Sidi Othmane, Casablanca 20023, Morocco*

**Abstract**

In response to the COVID-19 pandemic, a lot of scholarly articles have been published recently and made freely available. At the same time, there is emerging need to provide reliable and adequate information, which can reinforce the efforts made in raising awareness, to carry out infallible actions to prevent the worsening of the pandemic situation. With that said the current work tackles the perennial problem of creating a deep learning system allowing answering with a high precision to questions on a determined subject.

To do this, we will use open source scientific and academic articles concerning Covid-19, then we will proceed to the selection of suitable documents which will allow us to feed a question answering system based on BERT fine-tuned on the SQuAD benchmark.



*Keywords:* SQuAD ; BERT ; COVID-19 ; Question Answering

## 1. Introduction

* Corresponding author. Tel.: +212-657-478-632.
*E-mail address:* awanewidad93@gmail.com


10.1016/j.procs.2021.12.257

In recent years we have witnessed rapid progress in machine learning and in a more specific way in the field of linguistic understanding, with the introduction of several large-scale datasets, such as SQuAD (Rajpurkar et al., 2016), MS MARCO ( Nguyen et al., 2016), SearchQA (Dunn et al., 2017), TriviaQA (Joshi et al., 2017) and QUASAR-T (Dhingra et al., 2017). And we are witnessing at the same time the adoption of several Transformers [1], models to carry out text-classification tasks, next sentence prediction, QA...

It is argued that the integration of these two essential ingredients may allow us to witness an equivalent progress with regard to search engines, in this article we describe how the use of BERT, the powerful natural language processing model for general use, can allow us to create a search engine giving consistent and determined answers to the questions of the users, and for that we led our work on a database concerning the novel corona virus: covid-19, as an example.

## 2. System architecture overview

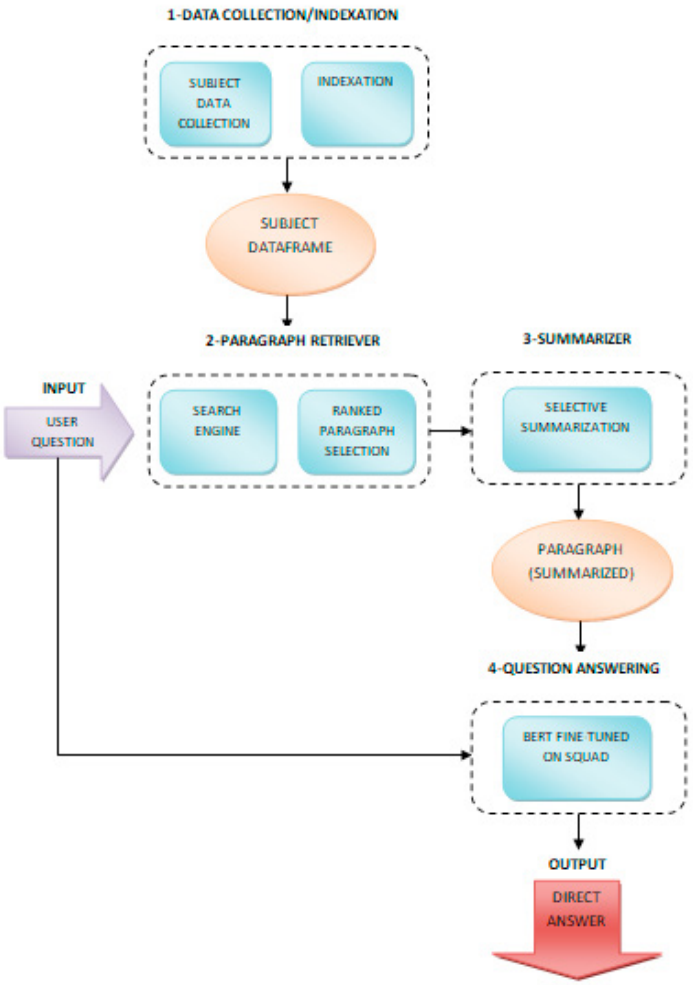


Fig. 1. System architecture

This diagram constitutes a description of the system designed in the form of a search engine, for a specific subject, in our case COVID-19, the process of said system can be detailed in several phases, namely:

DATA COLLECTION / INDEXING: In this step, we will collect a large database of scientific articles related to the COVID-19 topic, and also index the collected articles to create a database that will facilitate subsequent steps.

PARAGHRAPH RETERIEVER: In the second phase, it is a question of using the request of the user in order to constitute a selection of paragraphs which can contain the answer to the question entered beforehand, following this selection the choice will be made on the most significant paragraph through its score.

SUMMARIZER: At this level, we will apply a selective summarizer, which will allow us to select the most significant paragraphs in relation to the query.

QUESTION-ANSWERING: during this last step, the application of a BERT model fine tuned on the SQuAD benchmark will reveal the response by selecting the sentences considered as an answer to the query.

## 3. Data collection/indexation

### *3.1. Description of CORD-19*

In our case, the information will be collected by mining on a large open source Dataset created for scientific research purposes; we speak of the COVID-19 Open Research Dataset (CORD-19).

In response to the COVID-19 pandemic, the White House and a coalition of leading research groups prepared the COVID-19 Open Research Dataset (CORD-19).

CORD-19 is a resource of over 128,000 scholarly articles, including more than 59,000 in full text, on COVID-19, SARS-CoV-2 and related Coronaviruses. This freely available dataset is provided to the global research community to apply recent advances in natural language processing and other AI techniques to generate new perspectives in support of the ongoing fight against this infectious disease.

The dataset contains all research related to COVID-19 and Coronaviruses (eg SARS, MERS, etc.) from the following sources: PMM free access corp of PubMed using this request (COVID-19 search and coronavirus) Other COVID-19 research articles from a WHO-maintained corpus bioRxiv and medRxiv pre-prints using the same request as PMC (COVID-19 and coronavirus search) [2].

### *3.2. Indexation*

The texts we have are used in a first hand to populate various fields in a Dataframe that can be analyzed. Our analyse splits the text into small elements called tokens, the process known as tokenization. Multiple variants of analyzers that have different approach to creating tokens can be used, e.g. using different stemmers for extracting word roots. If the built-in Analyzers do not solve our problem we can combine built-in tokenizers and token filters or add our own Analyzer implementation.

After the documents have been analyzed they can be stored in the index structure. This is performed by IndexWriter, object that is responsible for creation and modification of the index. The particular index data structure used is inverted index [3]. The index called inverted because its keys are tokens that mapped to the documents in which they occur. The reason of using inverted indices is increased performance of finding the documents containing the token compared to forward indices. The index is stored in multiple segments with each segment being an index itself. Segments are created when index writes flushes its buffer and contain changed documents. Each segment is stored in a few files, responsible for different parts of the index. The segments will later be merged to improve index performance.

As an output of this phase we will have an index of thousands of scholarly articles allowing us to distinguish several areas of information: Researcher, Journal, Title, Abstract, Paragraphs etc...

### *3.3. Paragraph retriever*

In the present case and after having in our hands the index of the documents collected, we will use a search engine in the form of a tool-kit allowing us to search for the paragraphs corresponding to a question given by the user.

This work will be done using “Anserini” which is an open-source information search toolbox [4], built around “Lucene” to facilitate reproducible research.

Anserini provides envelopes and extensions on top of the basic Lucene libraries that allow researchers to use more intuitive APIs to perform common research tasks. As long as the results given for a question are sorted in a decreasing order of scores, the selection of the paragraph which will be used in the system of question answering, will correspond to the first paragraph appearing among the search results.

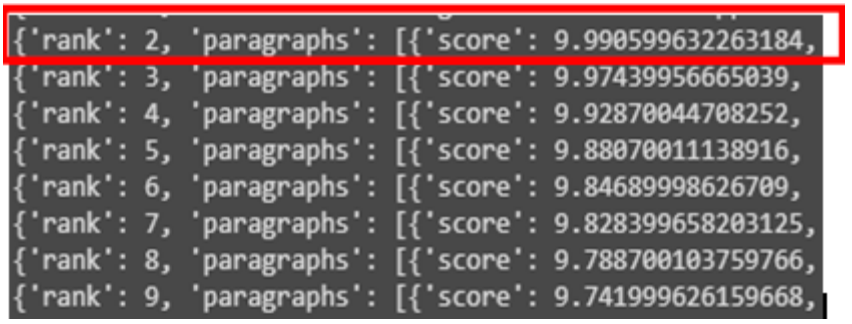


Fig. 2. Ranking of paragraphs in a search result

## 4. Summarizer

After our paragraph selection, a question response system can be applied directly to find a relevant answer to the user's question, but as we will work with a model handling a maximum size of 512 tokens for best results, a summary is an absolute must, to face the case where our selected paragraph exceeds the above-mentioned length. To carry out this task, two main options are offered:

-Extractive summarization: It is the strategy of concatenating extracts that are taken from a corpus into a summary. [5]

-Abstractive summarization: This one involves paraphrasing the corpus using novel sentences.

In our case, the balance tilts towards the extractive option, this is justified by the need to preserve the elements of the paragraph for use in the QA system, because the selection was made on the basis of the user's question, which will be used simultaneously in the upcoming phase. [6]

The tool that we will be using is a Bert based summarization model, called Bert Extractive Summarizer. This tool utilizes a transformers library to run extractive summarizations. This works by first embedding the sentences, then running a clustering algorithm, finding the sentences that are closest to the cluster's centroids. This library also uses coreference techniques to resolve words in summaries that need more contexts.

As arguments for our summarization model, we will set the minimum length of the summary to 60 tokens, so we want loose the meaningful parts, and its maximum to 512 as mentioned before to not exceed the QA models capacity.

## 5. Question-Answering

Our system for question answering will use the Bert model, fine-tuned on the SQuAD benchmark.
The BERT model was proposed in BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. [7]

It's a bidirectional transformer pre-trained using a combination of masked language modeling objective and next sentence prediction on a large corpus comprising the Toronto Book Corpus and Wikipedia, while Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles [8] , where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable.

This pre-trained transformer model has 24-layer, 1024-hidden, 16-heads, 340M parameters. [9 , 10]

## 6. Results

The results provided by this architecture are really impressive in terms of accuracy, which is, on one hand the

yield of using a fine-tuned model on SQuAD benchmark, whose results are correct up to 86%, and on the other hand the use of a widely enriched and updated database at the document collection level.

Table 1. Results analysis.

| RESULTS | EXISTING ANSWER IN ARTICLES | MISSING ANSWER IN ARTICLES | TOTAL |
|---|---|---|---|
| RIGHT ANSWERS | 71.25% | 0,00% | 71,25% |
| WRONG ANSWERS | 13,75% | 6,25% | 20,00% |
| NO ANSWERS | 3,75% | 5,00% | 8,75% |
| TOTAL | 88,75% | 11,25% | 100,00% |

*Examples:*


Fig. 3. Examples Question Answering

## 7. Conclusion

As we can see in the results, the use of a question answering system in integration with other different tools, such as the exploitation of indexing for the selection of documents from a large set of data, and the application of abstract summarizers to select specific paragraphs related to the query, can help us overcome the limitations of BERT models, where the size of the paragraph used to infer the answer to a question cannot exceed 512 characters, thus lead to a new generation of search engines, which has high success rates, exceeding 70% in revealing concise and targeted answers to our questions [8].

For further development we will try to implement Sci-BERT, a pre-trained BERT model to more accurately answer scientific questions.

That being said, the imperfections of such a system lie in its ability to understand natural language more broadly, which means that we still have a long lead over us in ML, in order to reach the level where machines can have capabilities similar to those of humans.

## References

[1] Jalammar.github.io/illustrated-transformer/ visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/ May 9, 2018.

[2] Lucy Lu Wang and Kyle Lo and Yoganand Chandrasekhar and Russell Reas and Jiangjiang Yang and Darrin Eide and Kathryn Funk and Rodney Michael Kinney and Ziyang Liu and William. Merrill and Paul Mooney and Dewey A. Murdick and Devvret Rishi and Jerry Sheehan

and Zhihong Shen and Brandon Stilson and Alex D. Wade and Kuansan Wang and Christopher Wilhelm and Boya Xie and Douglas M.Raymond and Daniel S. Weld and Oren Etzioni and Sebastian Kohlmeier CORD-19: The Covid-19 Open Research Dataset/Wang2020CORD19TC journal:ArXiv, volume:abs/2004.10706 .

[3] J. Wan and S. Pan, "Performance Evaluation of Compressed Inverted Index in Lucene," 2009 International Conference on Research Challenges in Computer Science, Shanghai, 2009, pp. 178-181, doi: 10.1109/ICRCCS.2009.53.

[4] Peilin Yang, Hui Fang, Jimmy Lin. Anserini: Enabling the Use of Lucene for Information Retrieval Research Computer Science ECIR 201

[5] Derek Miller, Leveraging BERT for Extractive Text Summarization on Lectures, arXiv:1906.04165, 2019.

[6] M. Ramina, N. Darnay, C. Ludbe and A. Dhruv, "Topic level summary generation using BERT induced Abstractive Summarization Model," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2020, pp. 747-752, doi: 10.1109/ICICCS48265.2020.9120997.

[7] Thilina Rajapakse, Question: How to use Transformers for Question Answering? Answer: Simple Transformers Novembre 17.2015.

[8] Zihuan Diao, Junjie Dong, and Jiaxing Geng. Question answering on squad dataset. CS224N 2018 Winter, 2018.

[9] Beliz Gunel and Cagan Alkan. Question answering on squad. CS224N 2018 Winter, 2018.

[10] Beliz Gunel and Cagan Alkan. Question answering on squad. CS224N 2018 Winter, 2018.