Question Answering System Using BERT Model in Python

A Project Report

Submitted to the Università di Pisa

in partial fulfillment of requirements for the award of degree

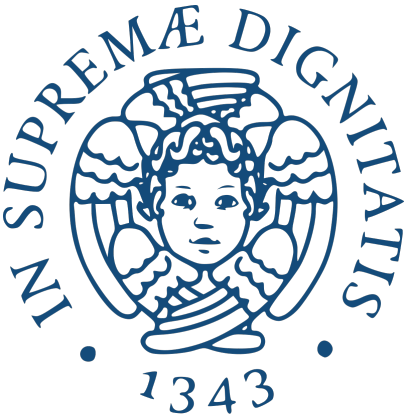
Master of Science

in

Data Science and Business Informatics

by

Amir Hassan(683185)



DEPARTMENT OF DATA SCIENCE AND BUSINESS INFORMATICS

UNIVERSITY OF PISA

PISA

Dec 2023

DEPARTMENT OF DATA SCIENCE & BUSINESS INFORMATICS

UNIVERSITY OF PISA

2023 - 24

CERTIFICATE

This is to certify that the report entitled **Question Answering System Using BERT Model in Python** submitted by **Amir Hassan** (683185) to the University of PISA in partial fulfillment of the MS. degree in Data Science and Business Informatics is a bonafide record of the project work carried out by him under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Prof. Laura Pollacci
(laura.pollacci [at] di [dot] unipi [dot] it)
Dept. of Computer Science
University of Pisa (UNIPI)
PISA

Dr. Antonio Frangioni
Professor and Head
Dept. of Data Science & Business Informatics
University of PISA
PISA

DECLARATION

We hereby declare that the project report **Question Answering System Using BERT Model in Python** , submitted for partial fulfillment of the requirements for the award of degree of MS University of PISA, PISA is a bonafide work done by us under supervision of Prof. Laura Pollacci

This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources.

We also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Pisa

1-11-2023

Amir Hassan

Abstract

This report investigates the creation and application of a Question Answering System (QAS) that makes use of the sophisticated features of the Python programming language's BERT (Bidirectional Encoder Representations from Transformers) and Word2Vec model. This study's main goal is to show how effective BERT is in understanding natural language, particularly when it comes to providing precise answers to queries that are asked in a particular context.

The methodology is all-inclusive and includes preprocessing, tokenization, setup, fine-tuning, and evaluation of the data. Utilizing the Transformers library from Hugging Face, the BERT model is trained on a dataset specifically designed for answering questions. The accuracy, precision, recall, F1-score, and other metrics are used to evaluate the model's performance.

Key findings demonstrate the BERT model's proficiency in natural language understanding by demonstrating its capacity to deliver accurate and contextually relevant replies to inquiries. The study goes over the finer points of parameter tuning, model training, and the effects of various setups on QAS performance.

This research concludes by highlighting the potential of BERT to transform Q&A systems and providing an analysis of its advantages, disadvantages, and future development directions for Python-based NLP applications.

Acknowledgement

I feel immense pleasure in expressing my deepest appreciation for the constant support and contributions of many individuals and those who provided us with the needed guidance, support, and motivation to achieve this milestone. First, thanks to Allah, the Almighty for His guidance throughout life. It's a matter of great pride and honor for us that we have been supervised by one of the best supervisors, Laura Pollacci. Indeed, without his guidance, cooperation, and motivating supervision, this work would have not been completed. We sincerely appreciate her valuable supervision, and support to us during the entire semester. Furthermore, we would also like to acknowledge with much appreciation the worthy president of the Data Science & Business Informatics, University of PISA, Dr. Antonio Frangioni, because without his sincerest efforts and wisest insights for the department, it would not have been possible to achieve this goal. We also appreciate the support of the staff of the Data Science Department, who made the resources available for us to utilize in completing this project. In In the end, we are also thankful to our well-wishers who supported and encouraged us to achieve our goals.

Amir Hassan

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The creation of Question Answering Systems (QAS) represents a major breakthrough in the fields of natural language processing (NLP) and information retrieval in the age of digital information. The goal of these systems is to close the gap that exists between human inquiries and the enormous textual knowledge sources.

1.1 Concept of Question Answering Systems (QAS)

Question Answering Systems (QASs) are a key advancement in natural language processing (NLP) that are intended to understand natural language questions and offer precise, contextually appropriate responses. In contrast to conventional keyword-based search engines, QAS uses advanced algorithms to decipher question meaning, comprehend question context, and extract accurate responses from textual data sources.

1.2 Significance of QAS in NLP and Information Retrieval

QAS are essential to the revolution in information retrieval because they allow users to engage with data in a way that is more natural and human-like. These systems improve user experience, productivity, and knowledge acquisition by enabling easy access to pertinent information from a variety of sources. Furthermore, because QAS offer quick access to precise information, they find use in a variety of fields, such as

research, education, healthcare, and customer service.

1.3 Role of BERT in Enhancing QAS

The capabilities of QAS have been greatly enhanced by the development of sophisticated machine learning models. Of them, the Bidirectional Encoder Representations from Transformers (BERT) model has attracted a lot of interest due to its profound comprehension of relationships and context in textual input. Pre-trained language representations and fine-tuning techniques from BERT have been crucial in helping QAS perform better in terms of accuracy and contextual comprehension, which in turn has improved its ability to retrieve correct answers from enormous corpora.

1.4 BERT Overview

BERT, which stands for Bidirectional Encoder Representations from Transformers, is a state-of-the-art natural language processing (NLP) model developed by Google. It revolutionized NLP tasks by introducing a bidirectional context understanding approach.

1.4.1 Architecture of BERT

BERT's architecture is based on Transformer models, utilizing a multi-layer bidirectional Transformer encoder. It consists of several self-attention layers that capture the relationships between words in both directions of a sentence. [1]

There are presently two versions available: BERT Base: 110 million parameters, 12 attention heads, and 12 layers (transformer blocks). BERT Large: 340 million parameters, 16 attention heads, and 24 layers (transformer blocks).

Bidirectional Encoder Representations from Transformers is the name of the BERT family of models, which processes each input text token in the whole context of all tokens before and after using the Transformer encoder architecture.

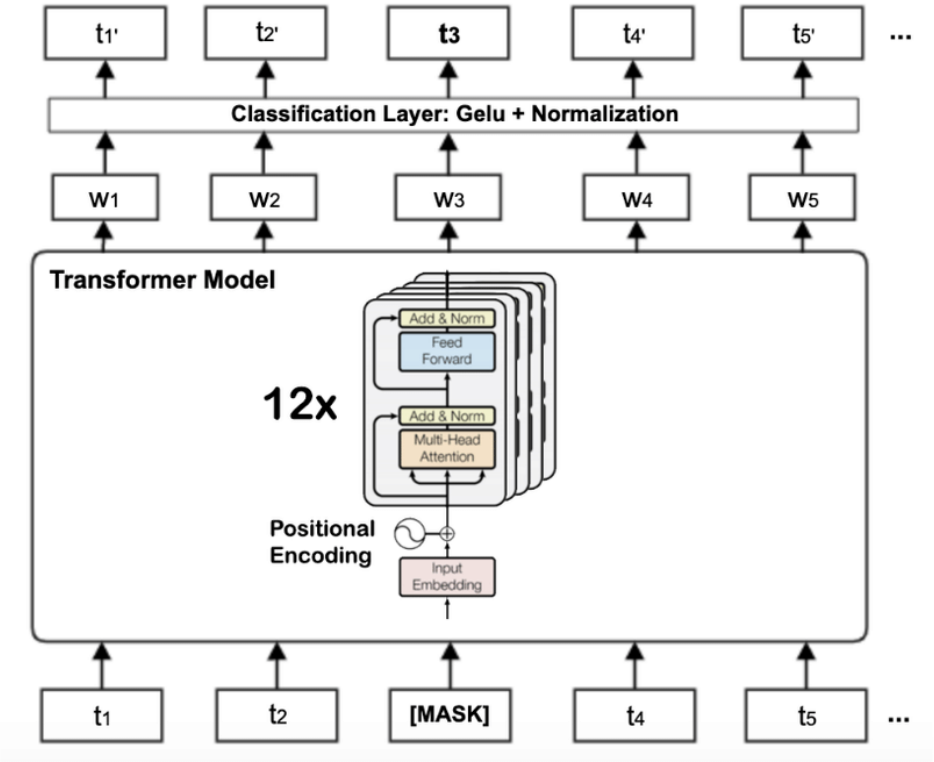


Figure 1.1: The-Transformer-based-BERT-base-architecture-with-twelve-encoder-blocks

Chapter 2

Literature Review

This chapter discusses similar applications and related work to our application. Every related work discussed here is highlighted with its description, major functionalities, and some limitations. [1]

2.1 Related work & Applications

[5] I have studied the systems mentioned below and after analyzing them with the best of our experience so far with this project, I highlighted some of the limitations and lack of helpful features in those systems.[6] Such as, I'm taking input from json, It shows the data through graphs and predict the answer from the dataset. [2]

2.2 Related Work

The QA system has been used to make various features such as chatbots, online QA, etc. In [1], question answering (QA) is framed by using reinforcement learning, and this approach is called Active Question Answering. The author suggested that between the user and a black box question-answering system, there is an agent that tries learning to reformulate queries to give the best possible answers. A brief study on Question Answering Systems is given in [2]. The paper effectively studies and provides research data on the different question-answering systems being used, the efficiency of the question-answering systems, and the areas where the Question-answering system can be used, which helps in answering and understanding the question-answering system.

[3] introduces one of the question-answering system methods, the BERT Model.' BERT stands for Bidirectional Encoder Representations from Transformers. The paper talks about how BERT is used as a question-answering model. It is effectively trained on a predefined dataset and then used to answer questions. It obtains good results on several natural language processing tasks. The paper talks about two important steps used in the framework: Pre-training the Bert and fine-tuning the Bert model. During pre-training, the training is done on the Bert model on the unlabeled data over various pre-trained tasks. [3]

Improvements in Question Answering using Natural Language Processing [4]. AnsweringOpen Domain factual questions use natural language processing to improve document selection and identify answers. [5] aims to develop an intelligent system that will learn from a text-based file and extract knowledge from the text of the given file. The system uses this extracted knowledge to answer queries input by the user. The main target of the Question Answering system (QAS) is to search in the systems, and it will return answers. The wide variety of users who want direct answers and large-scale evaluation of QA tasks benefit from QAS. In [6–7], the authors give a [4]detailed survey of the use of Named Entity Recognition to extract important information from Clinical texts, which is beneficial to be used in the biomedical field. The text features are extracted using different techniques for the applications in different domains for classification and information retrieval [8–10]. This system uses several proven and effective techniques to create an effective closed-domain QA system [11]. [5]

2.3 Similar Applications

[1]

Reference Shreya Acharya, K. Sornalakshmi, Bidisha Paul, Anshul Singh [India]
Summary The system will select the best probable answers from the stored database, which contains all the user answers using the NLP algorithm and give as an output. The data input will be given in the form of a text paragraph stored in a file.Firstly, whatever answer the user will be saved in a dataset inaCSV file.

Key Findings

1. NER techniques are applied from the dataset to retrieve the most important and

relevant data, which are highly unlikely to change in the long run.

2. Firstly, the CDQA method module is imported using python andthe BERT processor.

3. Checking the accuracy of the predicted model.

[2]

Reference Rana, Muhammad, [Bangladesh]

Summary This paper proposes to tackle Question Answering on a specific domain by developing a multi-tier system using three different types of data storage for storing answers. We compared different word and sentence embedding techniques for making a semantic question search engine and BERT sentence embedding gave us the best result.

Key Findings

1. Multi-Tier Question Answering System for Retrieving Answers From Heterogeneous Sources Using BERT. [6]

I) QA on Structured Data

II) QA on FAQ Data

III) QA on Unstructured Passage Data

[3]

Reference Chenxi Wang, Xudong Luo [China] **Summary** This proposes a scheme to obtain the problem vector representation based on the BERT model. In addition, the Milvus vector search engine is used, which can not only provide store vector representation information but also calculate vector similarity. Finally, we return the answer through the database.

Key Findings

The Q&A systems in the legal domain are very significant. In this paper, we develop a system of this kind. [7] Specifically, we first use the BERT Chinese pre-training model to obtain sentence vectors. Then we use the Milvus vector search engine to calculate the vector similarity.

Chapter 3

Methodology

This chapter highlights the objectives, design, architectural organization, and finally the access rights of users to the application. As mentioned earlier, the sole idea behind this project is to make easiness for people to get the exact answers.

The second most important objective was to facilitate not only the students but also their overall. Keeping these two prime objectives in mind we designed and developed carbon in two separate modules which are:

- Taking questions and predicting the precise answer accordingly.
- Showing the ratio through a graph and the accuracy of the model

These modules are then equipped with different AI as well as non-AI-based features to serve the needs of a particular user on the lower basis.

3.1 QA Pipeline

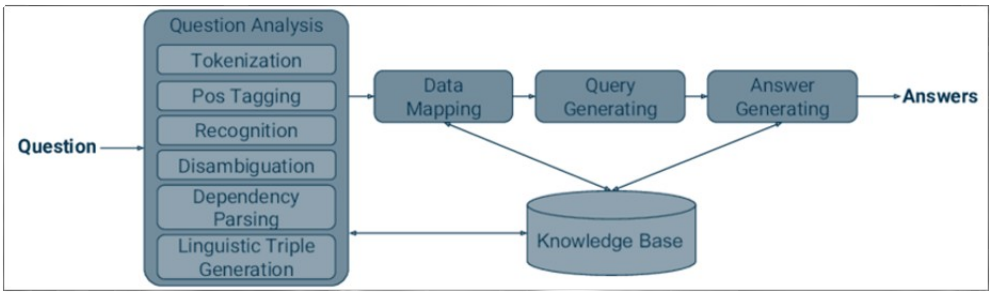


Figure 3.1: Pipeline for QA System

3.2 Architectural Design

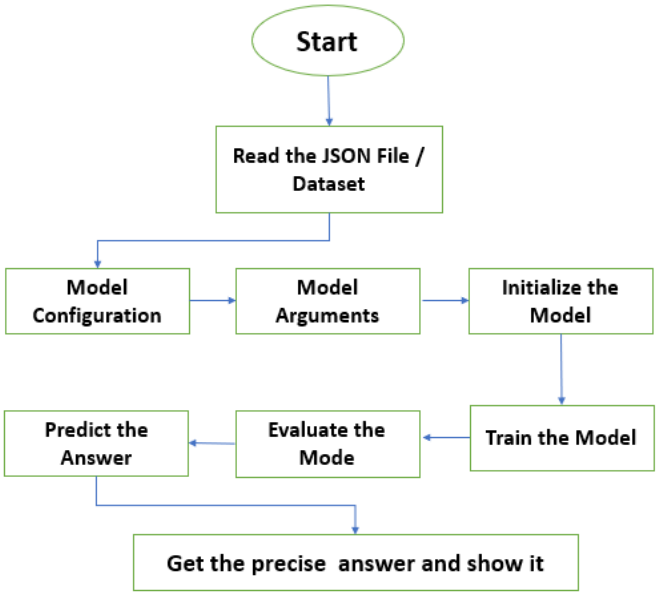


Figure 3.2: Flow Chart Diagram

Question Answering System Using BERT Model in Python

- Read Data From dataset File named as df
- Configure the Model
- Model Argument and Initialize it.
- Train the Model
- Evaluate the Model
- Make Predictions

Chapter 4

Implementations and Results

This project is developed to make a Question Answering System Using the Word2Vec and BERT Model in Python. As we have discussed earlier that concerned issue around the world getting the exact answer. Aim to make this QA system easily accessible and friendly to use.

4.1 Functional Implementation

Question Answering Using BERT (Bidirectional Encoder Representations from Transformers) and Word2Vec model has been build on python by using the Jypyter notebook on Anaconda.

Libraries

Installs the Hugging Face Transformers library

!pip3 install transformers

Installs the PyTorch library

!pip3 install torch

1- DataSet

Data Input: We use Question-Answer Dataset by Rachael Tatman as our dataset, which has the following attributes:

- ArticleTitle is the name of the Wikipedia article from which questions and answers initially came.

- Question is the question.

- Answer is the answer.
- DifficultyFromQuestioner is the prescribed difficulty rating for the question as given to the question-writer.
- DifficultyFromAnswerer is a difficulty rating assigned by the individual who evaluated and answered the question, which may differ from the difficulty in field 4.
- ArticleFile is the name of the file with the relevant article

The dataset is segmented into three sub-datasets, each corresponding to data collected in a different year. I begin by reading and merging the three subsets into a single dataframe.

2- Data Pre-processing Datapoints that contain NAN values in the ArticleFile or Answer columns are dropped from the dataframe. We then read the ArticleFile names for each datapoint, and extract the article text from that file into a new column ArticleText. Newlines (“”) in ArticleText are replaced with (“. ”) fullstops. The DifficultyFromQuestioner, DifficultyFromAnswerer, and ArticleFile are then dropped.

3- Data Cleaning The text in Question, Answer, and ArticleText columns is cleaned by removing ”[˙]”. The resultant text is converted into lower-case characters.

4- Embedding Models

Embedding models are based on co-occurrence and can extract the meaning of words using the contexts in which they appear. We leverage the ability of such models and adapt them to a question-answering scenario as our baseline models. Our implementations of embedding-based approaches are described in detail as follows:

1. Word2Vec-Based Question-Answering Model

I- Data Preprocessing: Convert the input data into a list of lists, presumably representing
- sentences or chunks of text.
- This data is then fed into the Word2Vec model.

II- Word2Vec Training:
- Train the Word2Vec model for 50 epochs.
- Use a fixed embedding size of 100 and a context window size of 8.

III- Question Answering Process:

- To answer a question, split the question into its component words.
- Pass these words into the Word2Vec model and generate embeddings for each word.
- Sum and average the generated embeddings to obtain an embedding for the entire question.
- Split the corresponding article text into individual sentences.
- Use a similar approach to find embeddings for each sentence in the article text.
- Calculate cosine similarity between the question embedding and each sentence embedding.
- Predict the sentence with the highest similarity as the answer to the given question.

Output:

The system outputs the selected sentence from the article text, which is deemed to have the highest Word2Vec embedding similarity with the given question.

By leveraging Word2Vec embeddings and cosine similarity, this simple question answering system aims to identify the most semantically similar sentence in the article text as the answer to a given question. It is noteworthy that while this approach provides reasonable accuracy, more sophisticated models, such as those based on deep learning or transformer architectures, might further enhance performance, especially on complex tasks or large datasets.

```
# Word2Vec outputs
print('Actual Question: ', my_question)
print('Answer without stopwords: ', rem_stop(my_question))
print('Actual Answer: ', df.iloc[index]['Answer'])
print('Euclidean distance: ', get_answer(my_question, sentences)) # Model's prediction using euclidean distance
# print("\n")
print('cosine similarity: ', get_answer_cosine(my_question, sentences)) # Our model's prediction using cosine similarity
```

```
Actual Question:  are there a large number of jews living in egypt today
Answer without stopwords:  large number jews living egypt today
Actual Answer:  no
Euclidean distance:   the oncevibrant jewish community in egypt has virtually disappeared with only a small number remaining in
the country but many egyptian jews visit on religious occasions and for tourism
cosine similarity:   the oncevibrant jewish community in egypt has virtually disappeared with only a small number remaining in
the country but many egyptian jews visit on religious occasions and for tourism
```

Figure 4.1: Word2Vec outputs

2. BERT Model Question Answering System

I- Model Architecture:

- Utilize BERT-based model, specifically BERTforQuestionAnswering, for question-answering tasks.

- Fine-tune the model on the SQuAD question-answer dataset, which provides a uniform representation for answers with consistent start and end positions.

II- Handling Large Answer Texts:

- Due to the 512-token limitation of BERT, split large answer texts into smaller chunks.

- Adopt Devlin et al.'s approach of using strides (512 tokens with a stride of 256 tokens) to reduce the risk of splitting answers across chunks.

III- Iterative Chunk Processing:

- Run the BERTforQuestionAnswering model iteratively on all chunks of the answer text.

- Collect potential answers, including beginning and ending indices (span) or [CLS] tag if no answer is found.

IV- Choosing the Best Answer:

- Experiment with different similarity measures between the question and the answer text to determine the best answer.

- Propose a method to select the best answer among multiple choices based on maximum similarity to the question text.

```
def getAnswerBert(question, context):

  # print('Query Context has {} tokens.'.format(len(tokenizer.encode(context))))

  context_list = get_split(context)

  ans = []

  for c in context_list:

    encoding = tokenizer.encode_plus(text=question,text_pair=c)

    inputs = encoding['input_ids']  #Token embeddings
    token_type_id = encoding['token_type_ids']  #Segment embeddings
    tokens = tokenizer.convert_ids_to_tokens(inputs) #input tokens

    output = model_bert(input_ids=torch.tensor([inputs]), token_type_ids=torch.tensor([token_type_id]))
    start_index = torch.argmax(output.start_logits)
    end_index = torch.argmax(output.end_logits)

    answer = ' '.join(tokens[start_index:end_index+1])

    ans.append(answer)
  print('Question: ', question)
```

Figure 4.2: Make predictions using the BERT model

V- Overall Model Performance: - The proposed BERT-based model is capable of handling large answer texts.

BERT Question Answering System Outputs

```
In [35]: print(getAnswerBert(df['Question'].iloc[666], df['ArticleText'].iloc[666]))
         print('Answer:', df['Answer'].iloc[666])

         Question:  What did James Monroe's letters not contain?
         [CLS] what did james monroe ' s letters not contain ? [SEP]
         Answer: No letters survive in which he might have discussed his religious beliefs.

In [36]: print(getAnswerBert(df['Question'].iloc[1222], df['ArticleText'].iloc[1222]))
         print('Answer:', df['Answer'].iloc[1222])

         Question:  What became one of the most important commercial and military centres of the British Empire?
         [CLS] what became one of the most important commercial and military centres of the british empire ? [SEP]
         Answer: Singapore

In [54]: print(getAnswerBert(df['Question'].iloc[555], df['ArticleText'].iloc[555]))
         print('Answer:', df['Answer'].iloc[555])

         Question:  was grover cleveland the twentyseventh president of the united states
         [CLS]
         Answer: no
```

Figure 4.3: BERT Question Answering

```
In [27]: print(getAnswerBert(df['Question'].iloc[234], df['ArticleText'].iloc[234]))
         print('Answer: ', df['Answer'].iloc[234])

         Question:  do all ducks quack
         very few ducks actually do qu ##ack
         Answer:  no

In [29]: print(getAnswerBert(df['Question'].iloc[192], df['ArticleText'].iloc[192]))
         print('Answer: ', df['Answer'].iloc[192])

         Question:  how has canada helped un peacekeeping efforts
         [CLS] how has canada helped un peacekeeping efforts [SEP]
         Answer:  during the suez crisis of 1956 lester b pearson eased tensions by proposing the inception of the united nations peacek
         eeping force canada has since served in 50 peacekeeping missions including every un peacekeeping effort until 1989

In [36]: print(getAnswerBert(df['Question'].iloc[1222], df['ArticleText'].iloc[1222]))
         print('Answer: ', df['Answer'].iloc[1222])

         Question:  what became one of the most important commercial and military centres of the british empire
         [CLS]
         Answer:  singapore
```

Figure 4.4: BERT Question Answering

Making answering prediction from the given context

```
In [35]: ### *********  Sample Questions picked up from the internet along with context ************

In [34]: # Sample user question picked up from the internet along with context
         getAnswerBert(('what does Amir do ?').lower(), ('Amir Hassan is Master student at University of in Data Science and Business inf

         Question:  what does amir do ?

Out[34]: 'web developer'

In [22]: # Sample user question picked up from the internet along with context
         getAnswerBert(('What does Laura Pollacci do at University of PISA?').lower(), ('Laura Pollacci is a PhD Student at the Computer

         Question:  what does laura pollacci do at university of pisa?

Out[22]: 'phd student'

In [23]:
         e studied arts and graphics at the Liceo Artistico Passaglia of Lucca in the Sperimental and Visual Ambiental course.').lower())

         Question:  what does laura pollacci teach at university of pisa?

Out[23]: 'text analytics'

In [24]: # Sample user question picked up from the internet along with context
         getAnswerBert(('Where is Laura Pollacci office at University of PISA?').lower(), ('Hi I am Laura Pollacci · room 288 @ Dpt. of C

         Question:  where is laura pollacci office at university of pisa?

Out[24]: 'room 288'
```

Figure 4.5: BERT Question Answering

Chapter 5

Conclusion and Future Work

In conclusion, there are several ways of building a question-answering system based on the input type, output type, and permitted complexity. This project allowed us to dive deeper into this area of machine learning and information retrieval and gave us hands-on experience with several modern ways of approaching this problem. We feel we are now confident to try out some of the latest ensemble models and even think of ways to apply our recently gained knowledge to try to make them work better. There is a lot more room for improvement and development in the future. Changes can be included in future work to improve it. [8] The BERT model is optimized as part of it. By utilizing an ML algorithm to make modifications to the CDQA system, the system may be enhanced and a more unique and comprehensible question-answering system can be created. [9]

As a future work, I would like to use the language models that come from the BERT language model in our future work. These models are the ALBERT (Lan et al., 2019) and RoBERTa (Y. Liu et al., 2019). Compared to the BERT model, which has superior performance, the RoBERTa model has been trained on more data and has produced a more efficient model. [10] Understanding of the spoken language. We can generate more reliable representations of the input sentences by using this approach. A lite BERT model is provided by the ALBERT model instead. The BERT model's performance has decreased as a result of this model's reduction in training parameters. [11] Rather, it enables us to use more intricate models in addition to the language model. Additionally, we can substitute more potent classifiers for the BERT model.

References

[1] M. C. W. G. A. G. N. H. e. a. C. Buck, J. Bulian, "Ask the right quest ions: Active quest ion reformulation with reinforcement learning," in *2022 IEEE Sensors*. IEEE, 2017, pp. 1–4.

[2] L. Kodra, Q. i. a. s. A. r. o. p. d. c. E. K. Meçe, and trends, "International journal of advanced computer science and applications," *ETSI white paper*, vol. 8, no. 10.14569, pp. 1–16, 2017.

[3] K. L. J. Devlin, M. W. Chang and K. Toutanova, "Bert: Pret raining of deep bidirectional transformers for language understanding," *ETSI white paper*, vol. 8, no. 1810.04805, pp. 1–16, 2018.

[4] A. L. KallirroiGeorgila TeruhisaMisu and D. Traum, "Reinforcement learning of question-answering dialogue policies for virtual museum guides," *ETSI white paper*, vol. 8, no. 10.14569, pp. 1–16, 2022.

[5] A. Singhal and D. Sharma, "New generalized 'useful' entropies using weightedquasi-linear mean for efficient networking," *ETSI white paper*, vol. 8, no. 10.14569, pp. 1–11, 2022.

[6] T. H. W. W. G. Zhou, Y. Zhou, "Learning semantic representation with neural networks for community question answering retrieval," *ETSI white paper*, vol. 93, no. 10.14569, p. 75–83, 2016.

[7] K. L. J. Devlin, M. W. Chang and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf*, vol. 1, no. 10.14569, pp. 41–46, 2019.

[8] X. Q. F. Sun, L. Li and Y. Liu, "U-net: Machine reading comprehension with unanswerable questions," *ETSI white paper*, vol. 3, 2018.

[9] L. P. Dinu and R. T. Ionescu, "A rank-based approach of cosine similarity with applications in automatic classification," *Algorithms Sci. Comput. SYNASC*, vol. 8, 2012.

[10] C. Google, "Xception: Deep learning with depthwise separable convolutions," *ETSI white paper*, vol. 8, pp. 1251–1258, 2014.

[11] H. J. Annamoradnejad I, Fazli M, "Predicting subjective features from questions on qa websites using bert. in: 2020 6th international conference on web research (icwr)," *ETSI white paper*, vol. 8, 2020.