Question Answering System using Deep Learning Approach

Victory Chimamaka Uwaoma
MSc Data Analytics
National College of Ireland
Dublin, Ireland
x19210931@student.ncirl.ie

Oyinlola Jadesola Popoola
MSc Data Analytics
National College of Ireland
Dublin, Ireland
x19138202@student.ncirl.ie

***Abstract*— In this paper, we explore how a pretrained model can be used for question answering using the Stanford Question Answering Dataset (SQuAD). This task of question answering benefitted from the transfer learning of the Bidirectional Encoder Representations from Transformers (BERT) natural language understanding architecture. The BERT model was pretrained by combining masked language modeling objective and next sentence prediction on a large corpus. The accuracy obtained is lower than the state of art but can be improved upon by further hyperparameter finetuning and using a large model instead of a base model.**

Keywords— natural language processing, question answering, transfer learning, BERT model

I. INTRODUCTION

Question answering has gained attention lately from information retrieval, natural language processing and machine learning [1]. The main purpose of a question answering machine is to extract the exact answer to questions. Information retrieval (IR) is a system that is set to return several materials containing information regarding the search query made to the system and this often times is time consuming especially when the materials returned are a lot [2] and this was traditional way of retrieving answers

Researchers in this field have only channelled their focus on natural language processing, knowledge base and information retrieval concepts [3]. The word information retrieval became known in 1951 by Calvin Mooers. Often times, these materials returned by the information retrieval systems may not contain the right information.

Motivated by the research about question answering systems that has recently gained recognition again since it came into lime light in the 1960s [4]. Unlike the traditional way of getting answers through information retrieval systems like WolframAlpha or IBM Watson, this question and answer system proposed using a pre-trained model will be able to retrieve the exact answer from a large passage that was created by [5]. The dataset known as SQuAD is a reading comprehension of a closed domain dataset that contains over a hundred thousand question-answer pair.

A. Research Question

This research project is aimed at applying deep learning techniques for question answering which will aid in getting answers to questions from paragraphs written in natural language. Hence will answer the research question:

To what extent can a Question Answering System built using pretrained models perform?

B. Objectives

The following objectives were put into consideration in order to meet the project goal and to address the research question.

a. The first objective was to carry out an extensive literature review on previous works carried out by researchers as well as the machine learning techniques they employed in building a question answering system.

b. The second objective involves data pre-processing on the dataset.

c. The third objective involves the implementation of the machine learning techniques chosen.

d. The fourth objective is to compare between the models produced and existing ones in the state of the art.

C. Structure of Thesis

The rest of the paper is structured accordingly:

Section II– Related Works: This section is the Related works talks about other question answering systems done by previous researcher.

Section III – Methodology: This section will talk about the implemented techniques that was used in building the question and answer system. Relevant diagrams, tables and figures will be shown.

Section IV– Evaluation and Results: This section presents the evaluation and results gotten from the model implemented in section III.

Section V– Conclusion and Future works: the research work is then concluded alongside future works that can be done in this field.

II. RELATED WORKS

A lot of research has been done in the area of question answering systems and this research is done to improve the current question answering systems. Using a deep learning models based on Long short-term memory (LSTM) to answer a question selection tasks, [6] proposed a model. This approach was divided into two segments, firstly using convolutional network alongside biLSTM (bidirectional long short-term memory) to accommodate questions and answers and using attention mechanism to generate answers

embeddings based on the question. Similarity metric was used to determine the closeness of the answer. The model did not rely on any feature engineering, outside supporting sources or language tools performed significantly well than other baseline models when tested on two different datasets; TREC-QA composed by Wang et al (2007) and InsuranceQA composed by (Feng et al., 2015), a non-factoid QA. The only limitation to this model is that the Future work seeks to further improve the model to perform different tasks, with regards to structure, the model will be improved to answer based phrase and sentential using attention mechanism and answer community QA. [7] used lexical semantic models based on WordNet to build an answer sentence selection for a question answering system. According to the authors, the paper was concentrated on the answer sentence selection from a given question and the response statement from people. The task is to find the correct phrase known to contain the exact answer and can reasonably support the choice of the answer. The model checks and pair words that are semantically related in order. Limitations of the word matching is that it involves misleading entity relationships, it requires high-level semantic representation and assumption and lack of comprehensive question analysis. The evaluated model was seen to outperform past works. In [8], the authors used supervised transfer learning for product information for a community question answering system. Using a transfer learning method on an existing Amazon community question answering dataset, a large volume of dataset which consist of over 3.1 million answers and 800,000 questions. The proposed model is set to map the questions to the similar product specifications which the baseline model will use as an input and output a score showing its importance. With the use of GloVe, a pre-trained word embedding, the given question Q and specification S are changed into two sequences $Q_e = [e^{Q}_{1}, e^{Q}_{2}, ..., e^{Q}_{m}]$ and $S_e = [e^{S}_{1}, e^{S}_{2}, ..., e^{S}_{n}]$. The sequences are then fed into biLSTM model as parameters where the sigmoid function outputs it as a probability between 0 and 1. The transfer learning is done in two steps, first, a large data source is used to train the pre-train the baseline model then fine-tuning is done on the target source (HomeDepotQA) of the same model. The target source is split into training, development and test dataset of 80%, 10%, 10% respectively. The AmazonCQA was preprocessed by deleting URLs containing questions or answers, limit the question length to four tokens, made answers to have a minimum of ten tokens, took out long questions and answers, took out phrase-like answers like “I have no idea” and sample negative answers. The model was then trained during the pre-training stage with improved hyperparameters and fine-tuning and the performance was checked using accuracy and mean reciprocal rank (MRR) metrics. After the creation of the model, it was then deployed as a mobile application intelligent shopping assistant (ISA) which helps enhance the shopping experience of shoppers in stores. The users can take a picture of the item in the store using the ISA app and it retrieves the information from the database which the users can further ask questions either by chatting in natural language or using voice. The model then gives the answer to the specification of the item being asked about by ranking the specification of the product based on its importance. In conclusion, the model proved to improve about 10% in terms of accuracy when using a large data source of existing community QA system.

Again, transfer learning was proposed by [9] was used in building a question answering system from SQuAD, a large supervised dataset. The aim of the model is to answer questions from a context, where the answer can be found in a sentence or a couple of sentences provided that is a context-aware QA pattern. Pre-training was done using the SQuAD data using BiDAF on both the source and target data. This takes the questions **q** and the paragraph **x** as the input and chooses the best answer using $\arg\max_{(i,j)} y^{start}_{i} y^{end}_{j}$, where $i <= j$. where y^{start}_{i} represents the beginning and y^{end}_{j} the end position. The authors also used BiDAF-T as another model, this was useful in the sentence-level. It takes the questions **q** and a set of sentences, $x_1,\ldots., x_T$ T represents the number of sentences. During the transfer learning, the weight of the target was set to that of the source model. Random number were used to transfer the weights from BiDAF on SQuAD to BiDAF-T. This was then evaluated on the two data source obtained, WikiQA [10] and SemEval 2016 (Task 3A) [11] which is slightly different from the SQuAD data. The results from the experiment showed that the WikiQA improved by 8% and SemEval by 1%. [12] introduced a general approach to designing a conservation question answering system. The aim is to show aspects of handling conservational history. They report that the approach involves three distinct elements: a BERT model which is integrated with a history model, and finally a history modeling part. The authors perform their experiments on the Question Answering in Context (QuAC) dataset and use F1 Score as the model’s performance metric.

The authors in [13] centered their work on building a question answering system using a pre-trained Bidirectional Encoder Representations from Transformers (BERT) model. The model had two steps pre-training and fine-tuning. Unlabeled data was used in the pre-training of the model and labeled data for fine-tuning. As compared to a model originally done by [14], the built proposed model was similar to that of the original model. Two BERT models were created with different parameters; $BERT_{BASE}$ (L=12, H=768, A=12, Total Parameters=110M) and $BERT_{LARGE}$(L=24, H=1024, A=16, Total Parameters=340M) where L represents the number layers, H the hidden layers and A the amount of self-attention heads. The pre-train BERT model was done using two unsupervised methods; Masked LM and Next Sentence Prediction (NSP). The later was executed by masking a percentage of random input token and predicting from the masked tokens which is then passed to an output softmax. The only disadvantage to this model is that it creates a mismatch between the fine-tuning and pre-training. The latter model was pre-trained for binarized next sentence prediction. During experiment, the GLUE was used in evaluating the model for its effectiveness on the SQuADv1 and v2 database produced by [5] and also on the SWAG (The Situations With Adversarial Generations) which has 113k sentences [15]. Feature-based approach was used with BERT during pre-training. The end result showed that the $BERT_{LARGE}$ is comparable with the state-of-the-art model using pre-trained transformers using both fine-tuning and feature-based approach. Researchers in [16] using natural language and by structured pruning of parameters from trained transformers to combine both the BERT- and RoBERTa-based question answering without any expensive pre-trained distillation attained better performance than the expensive pre-trained distillation models. The pruning approach used on the transformers utilized gate placement by masking each transformer. Every mask is gate variable represented in vectors as $\gamma_i \in [0, 1]$ and the mask is placed in every layer of a self-attention, Γ^{attn} of size $n_{H\ to}$ choose attention and every

feed-forward is also masked with Γ^{ff} of size d_l to choose an activations ReLU/GeLU using four methods in choosing the right gate values. SQuAD 2.0 and Natural questions were used for training of the gates. To avoid misuse of the dataset and for selecting a better hyper-parameter, 10% was used for testing the model while 90% was for training the gates. This experiment was done using the HuggingFace [17] while using a standard task-specific head with bert-base-uncased or bert-large-uncased. Results obtained from the SQuAD 2.0 model (large-qa) using a bert-large-uncased produced 86.4 F1 score on the dev dataset. The Natural questions dataset used transfer gates pruned BERT-large models with cross-entropy objective functions in its first phase. Distillation was combined with structured pruning when continuing with the training. The RoBERTa model when compared with the BERT-based model performed better in terms of accuracy because of the combination of structured pruning and distillation. Data augmentation approach was used in [18] for fine-tuning the BERT model with the same setup used by [19] who used a BERT-serini. The input to the model were pre-segmented paragraphs which is retrieved using BM25 ranking on the questions (bag of words) query to get the top paragraph *k*. The paragraphs and original questions in natural language for reference are then inputted into the BERT reader which uses the reference built with Google implementation with little changes which allows the correlation and aggregation of varying segments. The final layer of softmax was taken away over different answers spans [20]. The BERT-serini had a disadvantage according to [19] was that the fine-tuning was only done on the SQuAD. The experiment was performed on two datasets: SQuAD (v1.1) and TriviaQA [21] using 2016-12-21 dump of English Wikipedia by [22] for examination along with two other Chinese datasets with 2018-12-01 dump of Chinese Wikipedia. The performance metrics used included F1 score and exact match along with recall to compare with [19]. In conclusion, with data augmentation with fine-tuning improves the performance of the model.

The use of XLNet in building a question answering system was proposed by [23]. Baseline model such as BERT-base and BERT-large models were implemented using the AllenNLP [24] framework. The proposed XLNet imitates the Transformer(-XL) [25] rather than BERT [13]. Due to system constraint on a single GPU, the pre-processing carried out reduced the size of the context sequence of the dataset from 512 to 340 when fine-tuning on the GPU, but retains the 512 on the TPU (tensor processing unit). All text are changed to lowercase and tokenized using SentencePiece [26]. The experiment utilized twelve datasets; six in-domain and six-out domain. Multi-task learning based on pre-trained language model (MT-DNN) was used as proposed by [27] make used of nine natural language understanding (NLP) which performs better than the regular BERT models. In order to prevent overfitting, (MT-DNN) uses diverse loss functions on the different task by regularizing the language representation. In the end, the authors used multi-task to boost the generalization performance of a questing answering system by utilizing large pre-trained model on larger QA which achieved an exact match score (EM) of 56.59 and F1 score of 68.98 a little better than the BERT-large.

In [28], the authors present a multi-passage BERT model which normalizes answer scores among all word positions from all passages which correspond to the same question. This is done to address getting incomparable scores for answers which BERT models are trained by viewing passages that correspond with the same question independently. They report their approach outperforms the state-of-the-art and use EM and F1 Score as metrics for evaluation. Again, [29] used a six-module approach in tackling an open book QA system. The model used an OpenBookQA which has over 4900 questions and 500 multi-choice questions in both the test and validation. Using carefully top 10 selected knowledge facts and a BERT large model, the system was trained and knowledge was extracted. The model handled error analysis such as temporal reasoning, negation, conjunctive reasoning and qualitative reasoning. In conclusion, the system showed the downsides of using a BERT based MCQ models and had an accuracy of 72% which was 11.6% increase in performance from the state-of the art. Transfer learning was used by [30] to tackle QA system in a cost-effective way. A content-based QA system was utilized alongside retrieval information from three state-of-the-art module and five modules for answer extraction. Filter mechanism was put in use to customize the information retrieval. The customized information retrieval module used a Vector space model, Okapi BM25 model and Bigram model approach while the answer extraction module made use of embedding layers to change one-hot encoding to low dimensional space, encoding layers, question-context fusion which combines the question and context together and finally a prediction layer. Although, the prediction layer had a downside but was handled by BiDAF and R-Net.

Prior work [31] has suggested the used of transfer learning for question answering task. The SQuAD dataset used contains 100,000 comprehension questions [5]. Using the exact parameters as [22], the model was trained on DrQA system with a dimensional vector of 300 and glove embeddings of 840B. The experiment used two datasets coined from the SQuAD dataset; PLACES and WHO dataset. The data was tested on the development set after training. The pre-trained model was also used in training other focused datasets. The end result of the experiment based on exact match (EM) and F1 score showed that there was an increase of 3.31% in the F1 score and a 4.62% in the exact match. Other limitations of the model found were based on error analysis. Future work sees to implement transfer learning in Alquist socialbot to improve the bot. Another research carried out by [32] used BERT model in a question answering system for Italian e-invoicing [13]. The data was cleaned and fine-tuned leaving the data with 300 sentence pair containing a question and an answer. Half of the data was manually increased and generative grammars was used to increase the other half. The loss function puts two values into consideration neglecting the recall, accuracy and precision when training the data. The model was tested via several experiments. 88% accuracy was achieved when trained with other datasets. The model was compared with Google DialogFlow were it achieved an accuracy of 84% lower than the first. Future steps from the author seeks to improve the QA system in two ways; external and internal operations.

In [33], the authors used a simple approach using generative models for retrieving information in a question answering system. The authors used two methods in building the model. First the passage the passage was extracted before using sequence to sequence model in training. In the retrieval process, two approaches were used; BM25 [34]. Experiments done using three datasets focused on evaluation of Fusion-in-Decoder for the QA. With same parameters as s Lee et al. (2019), the model used 10% for evaluation and results gotten using exact match metrics by [5]**.** The proposed model was

compared with the state-of-the-art which performed significantly than the exiting models. Future work sets to improve the model to accommodate larger passages as well as implementing the retrieval model to learns end-to-end. The authors in [35] used a reasoning chain with BERT-based QA model for answer prediction on multi-hop questions [13]. WikiHop [36] and HotpotQA [37] datasets were used in the training of the model. The two chain extraction model; Oracle and uncased BERT, the model is trained to extract reasoning chains. The questions and context are taken as input and it outputs sequence of sentences. While getting the predicted answer, the chain extracted and the question is joined together with a pre-trained BERT model which is encoded. When evaluated, the WikiHop performed better than other existing models. The overall results shows the efficiency of using chains in a multi-hop question answering system.

Researches in [38] leverage the use of creating a supervised question based answering model using transfer learning in natural language. The transfer learning approach uses knowledge graph and seqtoseq model for meta-type identification and attribute level type identification. The authors also created alongside a curie framework to support the business analytics in interacting with the system in a user friendly manner. The curie framework can handle different queries such as factoid, ad-hoc, aggregations and transnational queries as well as responding using visualizations. Training and evaluation was done on the curie framework using SQL (structured query language) and CQL (cipher query language) in querying the database. The advantage of the model is its ability to work on cross-domain question answering system using deep neural networks. Even with less data, the model could still achieve promising results. Future models from the author seeks to integrate multi-tasking capabilities in the question answering model.

An effective approach using deep transfer learning was proposed by [39] in the medical domain for visual question answering system (Med-VQA). The use of ETM-Trans and co-attention mechanism approach was employed in this research with computer vision and natural language. The ETM-Trans was used as a feature extractor to enhance the features from images while using the co-attention mechanism with fusion techniques; Multimodal Factorized Bilinear (MFB) because of its ability to combine the visual features with the textual features. The model takes the question and the images as an input, processes it with the two approaches proposed where the co-attention mechanism uses a pre-trained ResNet152 model of the ImageNet and an LSTM to encode the questions to textual features and outputs the answer prediction through sampling method. All these steps were taking after pre-processing the dataset. The model was tested with the ImageCLEF 2018, the first model, ResNet152+ETM+MFH obtained the best result of 0.186 using WBSS score, ResNet152+MFH achieved the best result of 0.162 using BLEU score. [40] in their research proposed a unified Multi-task Knowledge Distillation Model (MKDM) for model compression used in multi-task framework. The model trains a student based on the knowledge gotten from training different teachers model. The MKDM uses the BERT model which has three layers [13]. Using the DeepQA dataset having questions from different domains was randomly sampled. After training the model, evaluation such as accuracy, queries per second and area under the curve were to check the models performance and compared with the baseline models. The model did significantly better than the baseline models. The end result showed that the model could learn generalized knowledge from different teachers using MKDM.

Lastly, [41] in their experiment used transfer learning framework on natural language (NL) and paraphrase identification (PI) queries through knowledge sharing. Different transfer learning and modelling techniques were evaluated and compared to the based models. Since the model proposed is a generalized model, any modelling used for sentence pair could adopt the model. Hybrid convolutional neural network (hCNN) model proposed is made up of two parts : SEbased BCNN model [42] and SI-based Pyramid model [43]. The BCNN model was modified to obtain a better performance while the pyramid used a matrix $M \in R^{m \times m}$ to show the similarity score using a dot product method. Intrinsic and extrinsic evaluation was done on the model to show the performance of the model. The proposed model using transfer learning performed better than past models.

III. Proposed Approach and Materials

This phase states explicitly the procedure taken to achieve the goal of the project. This procedure will adopt the Knowledge discovery in dataset approach (KDD) because it is majorly a scientific work. The Implementation of this work follows the KDD steps of Selection, Preprocessing Data, Transformation, Data Mining and Interpretations/ Evaluation. These steps are performed primarily using Python. The libraries and resources that support the development of the application are; pandas, numpy, seaborn gensim, matplotlib and scikit-learn. Fig 1 shows the structure of a Question Answering System (QA system).

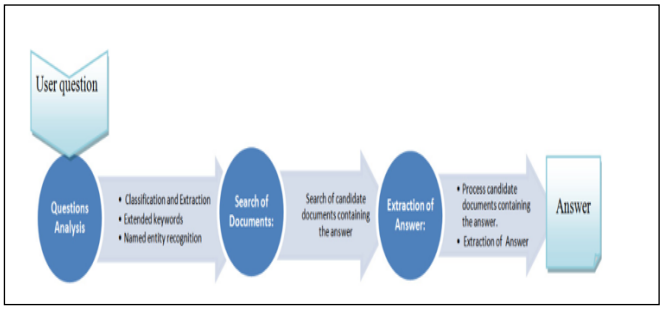

Figure 1: Structure of QA system (Bouziane, Bouchiha, Doumi and Malki, 2015) [44]

A. Dataset Selection

The dataset that is used in building the question and answering system was created by [5] and it is known as the Stanford Question Answering Dataset (SQuAD). It was generated by crowdworkers and it contains a reading comprehension that has more than a hundred thousand question-answer pairs based on more than five hundred articles. It is a closed domain dataset meaning that the answer to every question is a portion of text from the associated passage. A snippet of the dataset is shown below.

Checking Structure of the dataset

```
In [19]: train1.head()
```

Out[19]:

	index	question	context	answer_start	text	c_id
0	5733be284776f41900661182	To whom did the Virgin Mary allegedly appear i...	Architecturally, the school has a Catholic cha...	515	Saint Bernadette Soubirous	0
1	5733be284776f4190066117f	What is in front of the Notre Dame Main Building?	Architecturally, the school has a Catholic cha...	188	a copper statue of Christ	0
2	5733be284776f41900661180	The Basilica of the Sacred heart at Notre Dame...	Architecturally, the school has a Catholic cha...	279	the Main Building	0
3	5733be284776f41900661181	What is the Grotto at Notre Dame?	Architecturally, the school has a Catholic cha...	381	a Marian place of prayer and reflection	0
4	5733be284776f4190066117e	What sits on top of the Main Building at Notre...	Architecturally, the school has a Catholic cha...	92	a golden statue of the Virgin Mary	0

Figure 2: The table shows the snippet of the SQuAD dataset, alongside the ground-truth answer within it

B. Pre-Processing

The dataset is written in natural language processing (English) and must be pre-processed to be explored and used for its purpose. This involved checking the structure of the dataset to understand the features and attributes of the dataset. The datasets (training and dev) which originally exists as a very convoluted json file is untangled and converted into a clean dataframe.

The dataset had no null values therefore no action was required to deal with null values. Word Embeddings was done to examine the dataset for words with similar meanings. The cleaning process required before word embeddings involved word transformation to lowercase for uniformity, removal of punctuations, accent marks and other diacritics along with white space and redundant columns were also removed.

In this project, pre-trained transformer models using the simple transformers library from huggingface transformers was used[45] and in order to question answering in simple transformers, the data has to be preprocessed to be in the format of python list of dictionaries. Each dictionary has the "context" and "qas" (list of question and answers) attributes and the "qas" contains the id, a question and a list of correct answers. Figure 3 shows the conversion of the of the json file to the format suitable for using Transformers.

```
#converting the json files to list of dictionary which is the suitable format to perform QA in simple transformers
train = [item for topic in train['data'] for item in topic['paragraphs'] ]

type(train)
```

```
list
```

Figure 3: Conversion of train data to format suitable for transfomers.

C. Exploratory Data Analysis

Exploratory data analysis plays a big role in any data analysis problem and this is often done visually. It is mostly carried out to gain useful insights about the dataset.

1. Wordcloud

Wordcloud also known as tag cloud is a data visualization for text that shows the most frequent and important words in the dataset. Words of greater importance are written in bold, bigger fonts, whereas less significant words are displayed in smaller, lighter fonts or not displayed at all. The benefit of the Wordcloud is that it gives a quick peek at the text you are evaluating as this will help better in understanding dataset. The Wordcloud in this project was done in python using the pandas, matplotlib and Wordcloud packages.

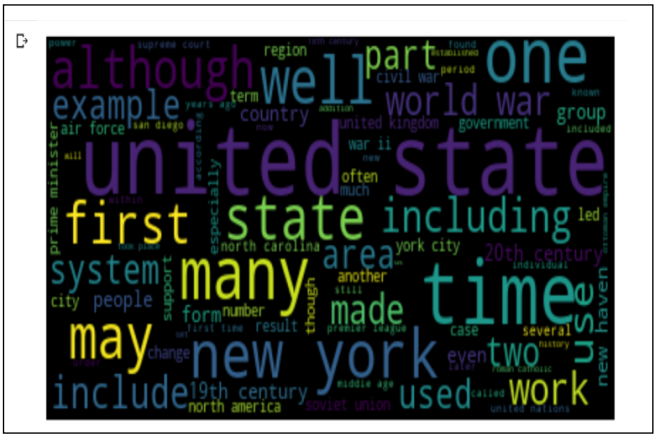

Figure 4:Word cloud for Context

The figure 4 above shows the visualization of the Wordcloud of context (paragraph) in the dataset. It show the frequency or the importance of word in different sizes.

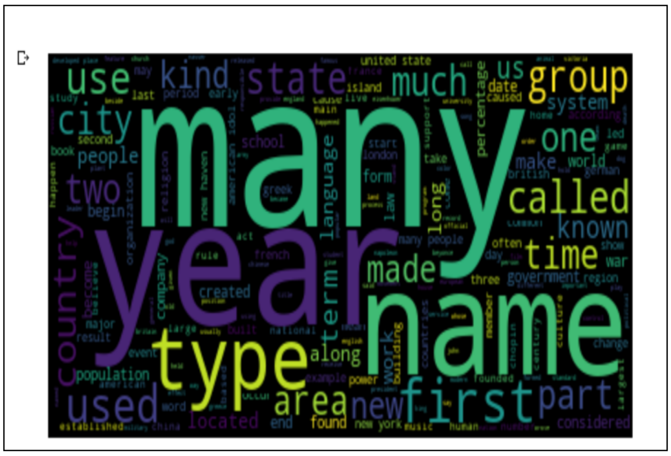

Figure 5: Word cloud for Questions

Figure 5 shows the visualization of the Wordcloud of questions in the dataset.

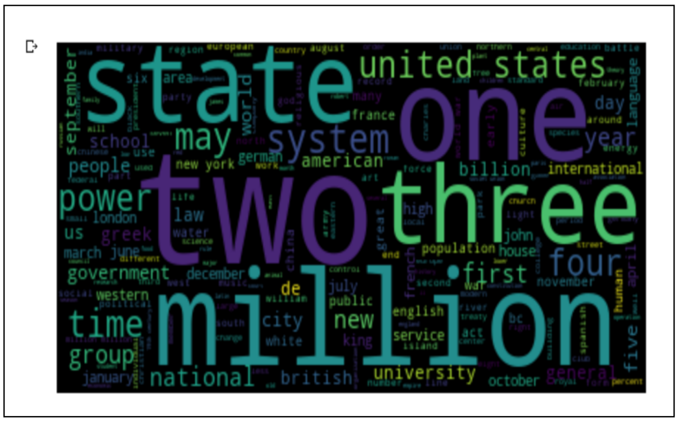

Figure 6: Word cloud for Text

Figure 6 shows the visualization of the Wordcloud of text (answers) in the dataset.

2. Word Embeddings

Word embeddings involves mapping words to vectors as this is the language computer understands, word embedding represents the words and sentences in several dimensions in vector space. It can be done in several ways but in this project, it was done using word2vec. Word Embeddings is done in this project to explore how similar words are to each other.

Word2vec

Word2vec is a technique that is used to perform word embedding with the use of neural network (Mikolov et al., 2013) which has an architectural structure of an input layer, hidden layer and output layer to model the words into vectors. Taking the paragraphs as input and output the words as vectors in the output layer. It has two methods in which embedding training can be carried out; continuous-bag-of-words (CBOW) and skip-gram. Word2vec uses the genism a python open-source library to implement word embedding. Similarity metric was used to check the closeness of words.

D. Application of Question answering System

The application of a question answering system is numerous. QA can be applied in frequently answered question (FAQ), as a chatbot for different organisations, as a web search such as google and encyclopaedia, for educational purpose.

E. Model

Transfer learning is an important machine learning methodology aimed at utilizing the information obtained from one task and transfer it to a new but similar task to either minimize the required fine-tuning, processing time or boost the accuracy of the model.

The use of Transfer learning is common in solving NLP problems including question answering especially with the use of pre-trained Transformer models. Transformer is a neural network architecture that was developed using self-attention mechanism that works very well when used for language understanding. In this project, the Simple Transformers library[46] which is built on top of Huggingface Transformers[45] is used to work with transformer models.

A class exist in simple transformers for each NLP task and the training, evaluation and prediction is carried out using an object of this class. The QuestionAnsweringModel object is created and the hyperparameters as shown in Figure 7 below for fine tuning the model is set. The compulsory parameters of the object are the model_type and model_name, while the args parameter is optional, if not provided default values are used.

The model_type supported by simple transformers includes BERT, ELECTRA, Longformer, XLM, ALBERT, DistilBERT, MobileBERT, XLM-RoBERTa, XLNet and RoBERTa. The BERT pre-trained model is used in this project for the question answering task.

The model_name specifies the architecture and trained weights to use. The 'bert-based-cased' is used. The 'bert-based-cased' is a Huggingface Transformers compatible pre-trained model which was pretrained in a self-supervised fashion on English language using masked language modelling and next sentence prediction[13]. BERT has given good results over time when used for NLP related tasks and it is presently used in Google search engine.

The BERT embeddings when compared to Word embeddings done using Word2Vec, gives different embeddings for the same words depending to the meaning in a context.

```
#using pre-trained transformer models using the simple transformers library from the huggingface transformers https://(
!pip install wandb
!pip install transformers
!pip install seqeval
!pip install tensorboardx
!pip install simpletransformers

from simpletransformers.question_answering import QuestionAnsweringModel


train_args = {
    'learning_rate': 3e-5,
    'num_train_epochs': 2,
    'max_seq_length': 384,
    'doc_stride': 128,
    'overwrite_output_dir': True,
    'reprocess_input_data': False,
    'train_batch_size': 2,
    'gradient_accumulation_steps': 8,
}

model = QuestionAnsweringModel('bert', 'bert-base-cased', args=train_args)
#bert is model type and bert-base-cased is the model name
```

Figure 7: The QuestionAnsweringModel object, hyperparameter values and configuration options

F. Implementation

Python programming language and Google Colab runtime environment is used for implementation. The transformers package provides an interface for using BERT.

A random sample of 5000 is taken from the train dataset to train the model. Also, a random sample of 1000 is also taken from the remainder of the train dataset to be used to evaluate the model. Random sampling was done due to the constraint of the system. The system configuration used was on Intel Core i5 with 2 GHz Quad-Core and 16GB memory, running a macOS. In taking the random sample, the function random.seed() is used in order to get the same sample every time which is for reproducibility. The dev dataset provided is not used to evaluate the model because it does not contain the ground truth therefore it can only be used for predictions. The train_model() method is used to train the model.

IV. Evaluation and Results

Fine tuning for 3 epochs with a learning rate of 5e-5 and train batch size of 32 is suggested in the BERT [13] paper but due to limited capability of available system, the model is trained using various hyperparameters and fine-tuned accordingly.

Using the eval_model() method to evaluate the model the metrics calculated are;

- the number of answers predicted that match the true answer correctly (defined as 'correct'),
- the number of answers predicted that are similar to the correct answer (defined as 'similar')
- the number of answers predicted that are neither correct nor similar (defined as 'incorrect')
- log loss of the data.

The eval_model object returns result and texts where result is a dictionary containing the evaluation results while texts contains the dictionary of correct answers, similar answers and incorrect answers.

The result returned after evaluating the 1000 random sample generated for testing is

- 2684 predicted answers match the true answers correctly.
- 1351 predicted answers are similar to the correct answer
- 622 answers are incorrect
- log loss of -8.06 and
- and an above average accuracy of 58% which can be improved upon if the dataset is large enough because BERT has been proved to perform better with very large dataset.

Predictions are made using the predict() method which returns;

- an answer list that contains question id mapped to its answer or a list of answers depending on the number of predictions set to be returned using the n_best_size parameter. This answer(s) given are top-k answers ranked by probability of answer.
- a list that contains question ids mapped to the probability score or a list of probability scores answers depending on the number of predictions set to be returned using the n_best_size parameter.

Performance evaluation based on three concrete examples that do not exist in the training dataset is done using the model.

In the first example, the context is- "*Linear regression is used for predicting quantitative values, such as an individual's salary. In order to predict qualitative values, such as whether a patient survives or dies, or whether the stock market increases or decreases, Fisher proposed linear discriminant analysis in 1936*", and the question is- "*Who proposed linear discriminant analysis?*". We let model give it top 2 answers where the model answers the question correctly and it is 99.9% sure of the answer.

In the second example, the context is- "*Thomas Alva Edison was an american inventor and businessman who has been described as America's greatest inventor. One of his inventions, is the phonograph*", the question is- "*Who invented the phonograph*?" and the model answers the question correctly and it is also very sure.

While in the third example, the context is- "*Mary drove for 3 hours to work in the morning and 5 hours in the evening to her house*" and the question is- "*How many hours did Mary drive for today?".* The model considers only the hours of driving in the morning and did not give the answer as 8 because it is not in the original text.

The prediction examples show the performance extent, exposes the limit and areas of future improvement of the BERT model.

V. Conclusion and Future works

The purpose of a question answering system is to extract answers to inquiries, unlike most information retrieval systems, rather than an answer, it retrieves an entire records or best matched excerpts from text. In this paper, we reviewed existing literature and methods used in QA systems and used the transformers architecture and pretrained models for question answering. It was noticed that the model would perform better with a larger dataset, therefore availability of larger datasets and computer systems with higher computing capability would give better results. Nevertheless, it can be concluded from the results gotten that;

- Question answering model can get used keywords in the question to get answers and it comprehends similar words (as seen in example 1 and 2)
- Reasoning of the model is quite poor if the relationship is complex (as seen in example 3)
- The model does not have mathematical capabilities such as summing up of numbers (as seen in example 3)

Due to the performance of the model, it is suitable to be applied in some industry cases but not too aggressive for other cases.

The capacity of future models of question answering systems would need to be improved upon to take advantage of digital technology already available. A mixture of natural language text, video, pictures, audio, tags just to mention a few. This will enable users express themselves using various ways to communicate their queries.

References

[1] E. Brill, S. Dumais, and M. Banko, 'An analysis of the AskMSR question-answering system', in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - EMNLP '02*, Not Known, 2002, vol. 10, pp. 257–264, doi: 10.3115/1118693.1118726.

[2] R. Gaizauskas and K. Humphreys, 'A Combined IR/NLP Approach to Question Answering Against Large Text Collections', p. 17.

[3] M. A. Calijorne Soares and F. S. Parreiras, 'A literature review on question answering techniques, paradigms and systems', *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 32, no. 6, pp. 635–646, Jul. 2020, doi: 10.1016/j.jksuci.2018.08.005.

[4] Z. Zheng, 'AnswerBus question answering system', in *Proceedings of the second international conference on Human Language Technology Research* -, San Diego, California, 2002, pp. 399–404, doi: 10.3115/1289189.1289238.

[5] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, 'SQuAD: 100,000+ Questions for Machine Comprehension of Text', *ArXiv160605250 Cs*, Oct. 2016, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/1606.05250.

[6] M. Tan, C. dos Santos, B. Xiang, and B. Zhou, 'LSTM-based Deep Learning Models for Non-factoid Answer Selection', *ArXiv151104108 Cs*, Mar. 2016, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/1511.04108.

[7] W. Yih, M.-W. Chang, C. Meek, and A. Pastusiak, 'Question Answering Using Enhanced Lexical Semantic Models', p. 11.

[8] T. M. Lai, T. Bui, N. Lipka, and S. Li, 'Supervised Transfer Learning for Product Information Question Answering', in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Orlando, FL, Dec. 2018, pp. 1109–1114, doi: 10.1109/ICMLA.2018.00180.

[9] S. Min, M. Seo, and H. Hajishirzi, 'Question Answering through Transfer Learning from Large Fine-grained Supervision Data', *ArXiv170202171 Cs*, Jun. 2018, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/1702.02171.

[10] Y. Yang, W. Yih, and C. Meek, 'WikiQA: A Challenge Dataset for Open-Domain Question Answering', in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015, pp. 2013–2018, doi: 10.18653/v1/D15-1237.

[11] P. Nakov *et al.*, 'SemEval-2017 Task 3: Community Question Answering', *ArXiv191200730 Cs*, Dec. 2019, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/1912.00730.

[12] C. Qu, L. Yang, M. Qiu, W. B. Croft, Y. Zhang, and M. Iyyer, 'BERT with History Answer Embedding for Conversational Question Answering', in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Paris France, Jul. 2019, pp. 1133–1136, doi: 10.1145/3331184.3331341.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding', in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, Jun. 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[14] A. Vaswani *et al.*, 'Attention is All you Need', p. 11.

[15] R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi, 'SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference', *ArXiv180805326 Cs*, Aug. 2018, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/1808.05326.

[16] J. S. McCarley, R. Chakravarti, and A. Sil, 'Structured Pruning of a BERT-based Question Answering Model', *ArXiv191006360 Cs*, Apr. 2020, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/1910.06360.

[17] T. Wolf *et al.*, 'HuggingFace's Transformers: State-of-the-art Natural Language Processing', *ArXiv191003771 Cs*, Jul. 2020, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/1910.03771.

[18] W. Yang, Y. Xie, L. Tan, K. Xiong, M. Li, and J. Lin, 'Data Augmentation for BERT Fine-Tuning in Open-Domain Question Answering', *ArXiv190406652 Cs*, Apr. 2019, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/1904.06652.

[19] W. Yang *et al.*, 'End-to-End Open-Domain Question Answering with BERTserini', *Proc. 2019 Conf. North*, pp. 72–77, 2019, doi: 10.18653/v1/N19-4013.

[20] C. Clark and M. Gardner, 'Simple and Effective Multi-Paragraph Reading Comprehension', *ArXiv171010723 Cs*, Nov. 2017, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/1710.10723.

[21] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, 'TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension', *ArXiv170503551 Cs*, May 2017, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/1705.03551.

[22] D. Chen, A. Fisch, J. Weston, and A. Bordes, 'Reading Wikipedia to Answer Open-Domain Questions', *ArXiv170400051 Cs*, Apr. 2017, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/1704.00051.

[23] D. Su *et al.*, 'Generalizing Question Answering System with Pre-trained Language Model Fine-tuning', in *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, Hong Kong, China, 2019, pp. 203–211, doi: 10.18653/v1/D19-5827.

[24] M. Gardner *et al.*, 'AllenNLP: A Deep Semantic Natural Language Processing Platform', *ArXiv180307640 Cs*, May 2018, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/1803.07640.

[25] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, 'Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context', *ArXiv190102860 Cs Stat*, Jun. 2019, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/1901.02860.

[26] T. Kudo and J. Richardson, 'SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing', *ArXiv180806226 Cs*, Aug. 2018, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/1808.06226.

[27] X. Liu, P. He, W. Chen, and J. Gao, 'Multi-Task Deep Neural Networks for Natural Language Understanding', *ArXiv190111504 Cs*, May 2019, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/1901.11504.

[28] Z. Wang, P. Ng, X. Ma, R. Nallapati, and B. Xiang, 'Multi-passage BERT: A Globally Normalized BERT Model for Open-domain Question Answering', *ArXiv190808167 Cs*, Oct. 2019, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/1908.08167.

[29] P. Banerjee, K. K. Pal, A. Mitra, and C. Baral, 'Careful Selection of Knowledge to solve Open Book Question Answering', *ArXiv190710738 Cs*, Jul. 2019, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/1907.10738.

[30] B. Kratzwald and S. Feuerriegel, 'Putting Question-Answering Systems into Practice: Transfer Learning for Efficient Domain Customization', *ACM Trans. Manag. Inf. Syst.*, vol. 9, no. 4, pp. 1–20, Mar. 2019, doi: 10.1145/3309706.

[31] J. Konrad, 'Transfer learning for question answering on SQuAD', p. 5, 2018.

[32] F. Alloatti, L. Di Caro, and G. Sportelli, 'Real Life Application of a Question Answering System Using BERT Language Model', in *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, Stockholm, Sweden, 2019, pp. 250–253, doi: 10.18653/v1/W19-5930.

[33] G. Izacard and E. Grave, 'Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering', *ArXiv200701282 Cs*, Jul. 2020, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/2007.01282.

[34] E. M. Keen, 'Okapi at TREC 3 S E Robertson S Walker S Jones M M Hancock-Beaulieu M Gatford Centre for Interactive Systems Research Department of Information Science City University Northampton Square London EC1V 0HB UK', p. 18.

[35] J. Chen, S. Lin, and G. Durrett, 'Multi-hop Question Answering via Reasoning Chains', *ArXiv191002610 Cs*, Oct. 2019, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/1910.02610.

[36] J. Welbl, P. Stenetorp, and S. Riedel, 'Constructing Datasets for Multi-hop Reading Comprehension Across Documents', *Trans. Assoc. Comput. Linguist.*, vol. 6, pp. 287–302, Dec. 2018, doi: 10.1162/tacl_a_00021.

[37] Z. Yang *et al.*, 'HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering', *ArXiv180909600 Cs*, Sep. 2018, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/1809.09600.

[38] L. H. U. and H. W. Lauw, Eds., *Trends and Applications in Knowledge Discovery and Data Mining: PAKDD 2019 Workshops, BDM, DLKT, LDRC, PAISI, WeL, Macau, China, April 14–17, 2019, Revised Selected Papers*, vol. 11607. Cham: Springer International Publishing, 2019.

[39] F. Crestani *et al.*, Eds., *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 10th International Conference of the CLEF Association, CLEF 2019, Lugano, Switzerland, September 9–12, 2019, Proceedings*, vol. 11696. Cham: Springer International Publishing, 2019.

[40] Z. Yang, L. Shou, M. Gong, W. Lin, and D. Jiang, 'Model Compression with Multi-Task Knowledge Distillation for Web-scale Question Answering System', *ArXiv190409636 Cs*, Apr. 2019, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/1904.09636.

[41] J. Yu *et al.*, 'Modelling Domain Relationships for Transfer Learning on Retrieval-based Question Answering Systems in E-commerce', in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining - WSDM '18*, Marina Del Rey, CA, USA, 2018, pp. 682–690, doi: 10.1145/3159652.3159685.

[42] W. Yin, H. Schütze, B. Xiang, and B. Zhou, 'ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs', *Trans. Assoc. Comput. Linguist.*, vol. 4, pp. 259–272, Dec. 2016, doi: 10.1162/tacl_a_00097.

[43] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, 'Text Matching as Image Recognition', *ArXiv160206359 Cs*, Feb. 2016, Accessed: Aug. 16, 2020. [Online]. Available: http://arxiv.org/abs/1602.06359.

[44] A. Bouziane, D. Bouchiha, N. Doumi, and M. Malki, 'Question Answering Systems: Survey and Trends', *Procedia Comput. Sci.*, vol. 73, pp. 366–375, 2015, doi: 10.1016/j.procs.2015.12.005.

[45] 'Hugging Face – On a mission to solve NLP, one commit at a time.' https://huggingface.co/ (accessed Aug. 12, 2020).

[46] T. Rajapakse, 'Simple Transformers', *Simple Transformers*. / (accessed Aug. 13, 2020).