P9: Automated query processing from passages Group No. 5

Gautam Sagar, 21CS60R15
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
West Bengal, IN
`gautamsagarnsit@kgpian.iitkgp.ac.in`

Yashas Rohan R, 19EE10066
Department of Electrical Engineering
Indian Institute of Technology Kharagpur
West Bengal, IN
`rohanryashas@gmail.com`

Razorshi Prozzwal Talukder, 21CS60A03
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
West Bengal, IN
`razorshi.prozzwal@kgpian.iitkgp.ac.in`

Vaibhav Vishal, 18EE10055
Department of Electrical Engineering
Indian Institute of Technology Kharagpur
West Bengal, IN
`rapbtycoon256@gmail.com`

Abstract

Knowledge base question answering (KBQA) is an important task in Natural Language Processing. Question Answering (QA) systems enable users to retrieve exact answers for questions posed in natural language.Question Answering (QA) over Knowledge Base (KB) aims to automatically answer natural language questions via well-structured relation information between entities stored in knowledge bases.In this project we aim to implement traditional and modern State of the Art question answering system over a long passage. The Passage chosen is taken from a history text book. The length of the passage chosen is more than 7000 words. This project demonstrates the ability of traditional QA systems and modern Deep learning based QA systems.

1 INTRODUCTION

Question Answering (QA) systems have emerged as powerful platforms for automatically answering questions asked by humans in natural language using either a pre-structured database or a collection of natural language documents. In other words, QA systems make it possible asking questions and retrieve the answers using natural language queries and may be considered as an advanced form of Information Retrieval (IR). Knowledge Base QA Find answers from structured data source (a knowledge base) instead of unstructured text. Standard database queries are used in replacement of word-based searches. This paradigm, make use of structured data, such as ontology. An ontology describes a conceptual representation of concepts and their relationships within a specific domain. Ontology can be considered as a knowledge base which has a more sophisticated form than a relational database. The fundamental problem for retrieval-based QA systems is to retrieve the most similar question from the QA knowledge base given a query, so as to provide the respective answer. Such a text (i.e., query-question) matching problem can be represented as Paraphrase Identification (PI) or some form of Natural Language Inference (NLI).When dealing with such text matching problems in the real world industrial-scale QA applications, we are facing two prominent challenges, i.e., i) the lack of abundant data to learn a model with high accuracy and ii) the requirement of high inference speed for online model serving. Recent advances on text matching rely heavily on the flourishment of deep learning models. On the one hand, those deep models are proven to be effective when rich in-domain labeled data is available. However, in real-world applications, it is challenging to obtain a sufficient amount of labeled data for every domain of interest, as data annotation is commonly time-consuming and costly. On the other hand, high Query-Per- Second (QPS) requirements for seamless online serving demand the deployed models to be light-weight. Thus the trained models have to be either designed to be simple in structure but effective in performance, or compressed if well-performed large models are originally trained. With recent emergence if language models like BERT which have Billions of parameters the main question that arises is how do we utilize pre-trained models together with other data sources from different domains to facilitate knowledge transfer that is effective in performance and efficient in serving?

In this project we have implemented BERT based question answering system which takes in questions and a passage supplied by the user and answer the question. One traditional method of QA is also demonstrated and finally both the methods are compared.

2 CASE STUDY

2.1 Modern SoTA Methods

This section presents a short case study in three recent state-of-the-art language models that have presented state-of-the-art results in many NLP tasks.

2.1.1 BERT(Bidirectional Encoder Representations from Transformers)

BERT makes use of Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text. In its vanilla form, Transformer includes two separate mechanisms — an encoder that reads the text input and a decoder that produces a prediction for the task. Since BERT's goal is to generate a language model, only the encoder mechanism is necessary.

As opposed to directional models, which read the text input sequentially (left-to-right or right-to-left), the Transformer encoder reads the entire sequence of words at once. Therefore it is considered bidirectional, though it would be more accurate to say that it's non-directional. This characteristic allows the model to learn the context of a word based on all of its surroundings (left and right of the word).

When training language models, there is a challenge of defining a prediction goal. Many models predict the next word in a sequence, a directional approach which inherently limits context learning. To overcome this challenge, BERT uses two training strategies:

Masked LM (MLM)

Before feeding word sequences into BERT, 15% of the words in each sequence are replaced with a [MASK] token. The model then attempts to predict the original value of the masked words, based on the context provided by the other, non-masked, words in the sequence. The BERT loss function takes into consideration only the prediction of the masked values and ignores the prediction of the non-masked words. As a consequence, the model converges slower than directional models, a characteristic which is offset by its increased context awareness.

Next Sentence Prediction (NSP)

In the BERT training process, the model receives pairs of sentences as input and learns to predict if the second sentence in the pair is the subsequent sentence in the original document. During training, 50% of the inputs are a pair in which the second sentence is the subsequent sentence in the original document, while in the other 50% a random sentence from the corpus is chosen as the second sentence. The assumption is that the random sentence will be disconnected from the first sentence.

2.1.2 XL-Net

XLNet is a BERT-like model which is a breakthrough in NLP as it outperforms the state-of-the art BERT algorithm in 20 different tasks.

- BERT is categorized as Auto Encoder(AE) language model which aims to reconstruct the original data from corrupted input. The advantages of AE language model is that it can see the context on both forward and backward direction. But the AE language model also has its disadvantages. It uses the [MASK] in the pretraining, but this kind of artificial symbols are absent from the real data at finetuning time, resulting in a pretrain-finetune discrepancy. Another disadvantage of [MASK] is that it assumes the predicted (masked) tokens are independent of each other given the unmasked tokens.
- XLNet is categorized as Auto Regressive(AR) language model which uses the context word to predict the next word. But here the context word is constrained to two directions, either forward or backward. The advantages of AR language model are good at generative NLP tasks. Because when generating context, usually is the forward direction. AR language model naturally works well on such NLP tasks. But AR language model has some disadvantages, it only can use forward context or backward context, which means it can't use forward and backward context at the same time.

XLNet propose a new way to let the AR language model learn from bi-directional context to avoid the disadvantages brought by the MASK method in AE language model.

Language model consists of two phases, the pre-train phase, and fine-tune phase. XLNet focus on pre-train phase. In the pre-train phase, it proposed a new objective called Permutation Language Modeling. We can know the basic idea from this name, it uses permutation. AR language model only can use the context either forward or backward, so this way lets it learn from bi-directional context.

2.1.3 ALBERT

A Lite BERT, nicely abbreviated to ALBERT is created by changing a few things in BERT's architecture, capable of achieving the same performance as BERT, but only at a fraction of the parameters and hence computational cost. This overcomes one of the biggest drawbacks of BERT. It is changed in three key ways, which bring about a significant reduction in parameters:

- Embeddings are factorized, decomposing the parameters of embedding into two smaller matrices in addition to adaptations to embedding size and hidden state size.
- Applying cross-layer parameter sharing. In other words, parameters between certain subsegments from the (stacked) encoder segments are shared, e.g. the parameters of the Multi-head Self-Attention Segment and the Feedforward Segment. This is counter to BERT, which allows these segments to have their own parameters.
- Following post-BERT works which suggest that the Next Sentence Prediction (NSP) task utilized by BERT actually underperforms compared to what the model should be capable of, Lam et al. (2019) introduce a sentence-order prediction (SOP) loss task that actually learns about sentence coherence.

Table 1: The configurations of the main BERT and ALBERT models analyzed in the original ALBERT paper.

Model	Parameters	Layers	Hidden State Size	Embedding Size	Parameter sharing
BERT	base	108M	768	768	False
	large	334M	1024	124	False
	xlarge	1270M	2048	2048	False
ALBERT	base	12M	768	128	True
	large	18M	1024	128	True
	xlarge	59M	2048	128	True
	xlarge	233M	4096	128	True

2.2 Traditional QA System

2.2.1 LUNAR System

The LUNAR (QA) System is a Question Answering System that was first demonstrated at a lunar science convention in 1971. It interfaced with an Expert System (Expertise Domain) in lunar rock

samples. It used an Augmented Finite-State Transition Network grammar (ATN) parser combined with a rule-based semantic interpretation program to guided its question analysis. The LUNAR system allowed a user to ask questions, compute averages and ratios, and make listings of selected subsets of the data. One could also retrieve references from a keyphrase index and make changes to the data base. The system permitted the user to easily compare the measurements of different researchers, compare the concentrations of elements or isotopes in different types of samples or in different phases of a sample, compute averages over various classes of samples, compute ratios of two constituents of a sample, etc., all in straight forward natural English.
The LUNAR system consists of three principal components: a general purpose grammar and parser for a large subset of natural English, a rule driven semantic interpretation component using pattern + action rules for transforming a syntactic representation of an input sentence into a representation of what it means, and a database retrieval and inference component that stores and manipulates the data base and performs computations on it. The first two components constitute a language understanding component that transforms an input English sentence into a disposable program for carrying out its intent (answering a question or making some change to the data base). The third component executes such programs against the data base to determine the answer to queries and to effect changes in the data base.
The system contains a dictionary of approximately 3500 words, a grammar for a fairly extensive subset of natural English, and two data bases: a table of chemical analyses with 13,000 entries, and a topic index to documents with approximately 10,000 postings. The system also contains facilities for morphological analysis of regularly inflected words, for maintaining a discourse directory of possible antecedents for pronouns and other anaphoric expressions, and for determining how much and what information to display in response to a request.
The grammar used by the parsing component of the system is an augmented transition network (ATN).

2.2.2 BASEBALL system

Baseball is a computer program that answers questions phrased in ordinary English about stored data first demonstrated in 1961. The program reads the question from punched cards. After the words and idioms are looked up in a dictionary, the phrase structure and other syntactic facts are determined for a content analysis, which lists attribute-value pairs specifying the information given and the information requested. The requested information is then extracted from the data matching the specifications, and any necessary processing is done. Finally, the answer is printed. The program's present context is baseball games; it answers such questions as "Where did each team play on July 7?". It is a computer program that answers questions posed in ordinary English about data in its store. The program consists of two parts. The linguistic part reads the question from a punched card, analyzes it syntactically, and determines what information is given about the data being requested. The processor searches through the data for the appropriate information, processes the results of the search, and prints the answer. The program is written in IPL-V , an information processing language that uses lists, and hierarchies of lists, called list structures, to represent information. Both the data and the dictionary are list structures, in which items of information are expressed as attribute-value pairs, e.g., Team = Red Sox.
The program operates in the context of baseball data. At present, the data are the month, day, place, teams and scores for each game in the American League for one year. In this limited context, a small vocabulary is sufficient, the data are simple, and the subject-matter is familiar. Some temporary restrictions were placed on the input questions so that the initial program could be relatively straightforward. Questions are limited to a single clause; by prohibiting structures with dependent clauses the syntactic analysis is considerably simplified. Logical connectives, such as and, or, and not, are prohibited, as are constructions implying relations like most and highest. Finally, questions involving sequential facts, such as "Did the Red Sox ever win six games in a row?" are prohibited. These restrictions are temporary expedients that will be removed in later versions of the program. Moreover, they do not seriously reduce the number of questions that the program is capable of answering. From simple questions such as "Who did the Red Sox lose to on July 5?" to complex questions such as "Did every team play at least once in each park in each month?" lies a vast number of answerable questions.

3 BERT based QA

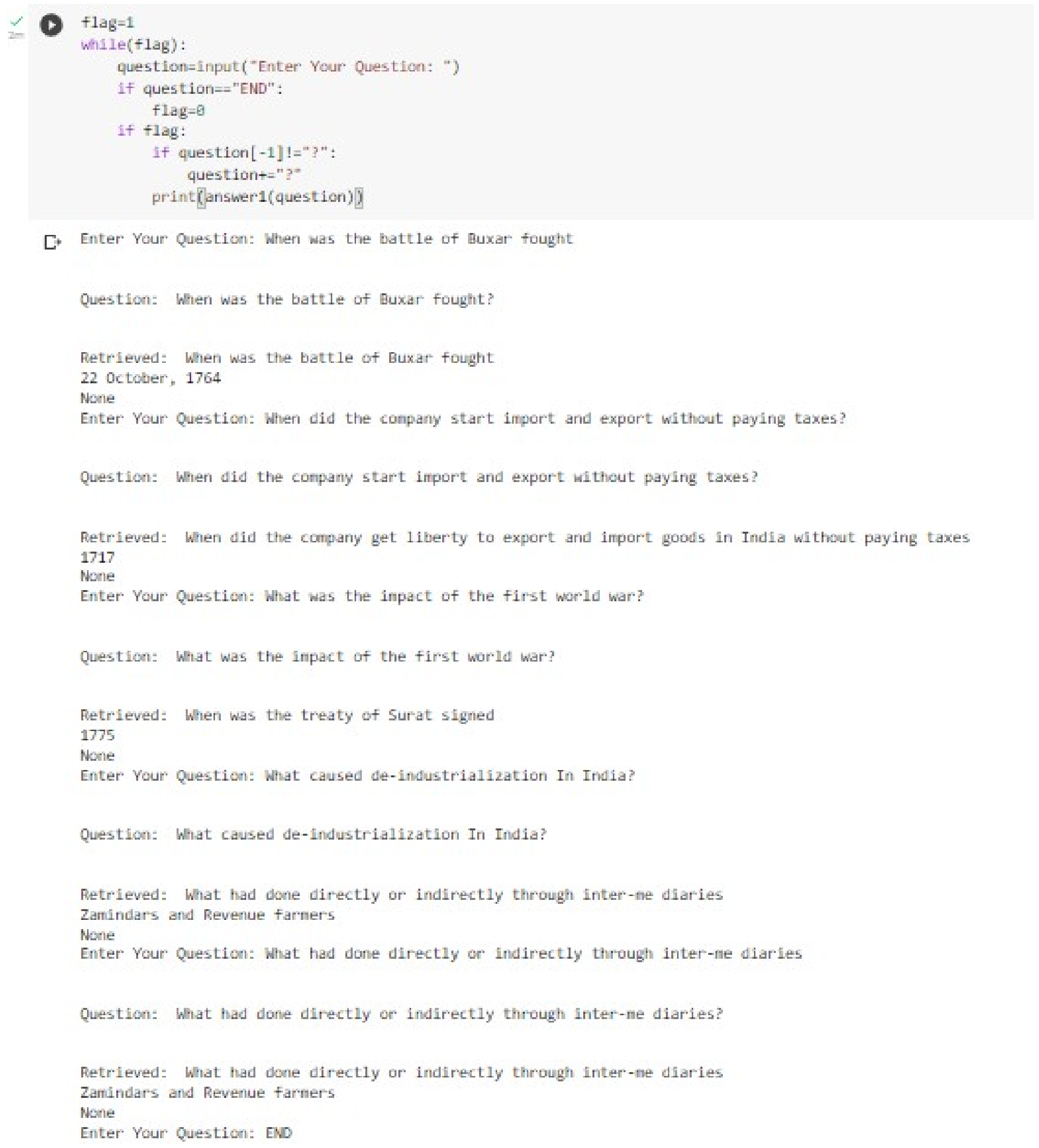
To implement BERT based QA we have used the HuggingFace library. Transformer is their NLP Library that contains state-of-the-art transformer models. To feed question and answer into the BERT model the two pieces of text i.e. question and the answer are joined by [SEP] token. This combined text is then encoded into appropriate form. After input tokenization and encoding we get input encoding in which each token is attached with a token id. From the HuggingFace model for BERT question answering we get start and end token index of the answer present in the passage. Score for these answers are calculated separately. Based on the start and end index we construct the answer from the passage and return it. The in-built model has a limitation of 512 tokens at the input so we have to break the passage into 500 tokens and slide the window of 500 tokens by 250 tokens at each step. This increases the inference time for the final answer but it also increases the accuracy of the answer. The answer with highest score is chosen as the final answer. To use our implementation, a user has to enter a question and wait for the answer. Final answer may appear after 3-4 min.. To end the program user will have enter "END" in the question prompt. User also has the option to save his question in a text file named as "questions.txt" and uncomment the appropriate portion of the code which is also indicated in the notebook. The passage is always supplied to the program in the form of a text file named as "Passage.txt". Some examples are present in the notebook itself.

4 Traditional QA System

In the traditional systems for question answering (rule-based), we use word2vec, glove and bag-of-word models to convert the given question into a query vector (or phrase embedding). We have made a database of approximately a 100 questions spanning the entire passage in the study, consisting of both objective and subjective questions. The formed query vector uses cosine similarity to retrieve the answer to the closest matching question available in the database and provides an answer. It can already be seen that the performance is directly based upon the size of our question-answer database and sometimes, this can be very daunting and expensive to prepare because for the best accuracy, the database should be made manually. Also if the question asked isn't present in our database, a similar question is picked up and an answer to the latter is given which may be completely out of context to the question asked. In the following section, we perform a study on the performance of these models that create word embeddings which are then converted to phrase embeddings.

4.1 Bag of Words

Bag of words is a relatively simple method in which a vector is generated which is a dictionary representation of the given question in which each word has the number of times it appeared in the question. For this, first a vocabulary of all the words in the database questions is created and the model then creates the embedding with the help of the said vocabulary. There are no hyperparameters involved, the only parameter which will change is the size of our vocabulary. A vocabulary of size 281 is created, using this, every question in the database is embedded into a vector. Now the question asked by the user is also converted into this format and cosine-similarity from sklearn is used to retrieve the answer to the most similar question. Here for our testing purposes,we have randomly prepared 5 questions from the passage, and queried them to check the performances of each model on the same.



```
flag=1
while(flag):
    question=input("Enter Your Question: ")
    if question=="END":
        flag=0
    if flag:
        if question[-1]!="?":
            question+="?"
        print(answer1(question))
```

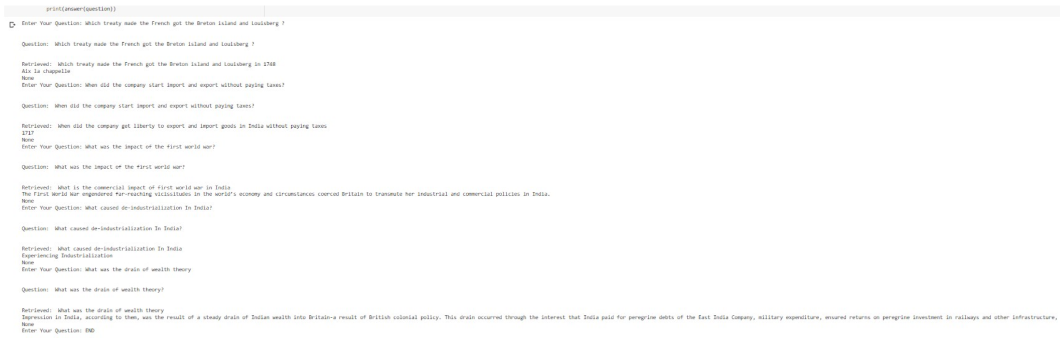
```
Enter Your Question: When was the battle of Buxar fought


Question:  When was the battle of Buxar fought?


Retrieved:  When was the battle of Buxar fought
22 October, 1764
None
Enter Your Question: When did the company start import and export without paying taxes?


Question:  When did the company start import and export without paying taxes?


Retrieved:  When did the company get liberty to export and import goods in India without paying taxes
1717
None
Enter Your Question: What was the impact of the first world war?


Question:  What was the impact of the first world war?


Retrieved:  When was the treaty of Surat signed
1775
None
Enter Your Question: What caused de-industrialization In India?


Question:  What caused de-industrialization In India?


Retrieved:  What had done directly or indirectly through inter-me diaries
Zamindars and Revenue farmers
None
Enter Your Question: What had done directly or indirectly through inter-me diaries


Question:  What had done directly or indirectly through inter-me diaries?


Retrieved:  What had done directly or indirectly through inter-me diaries
Zamindars and Revenue farmers
None
Enter Your Question: END
```

We can see that it got 3 out of 5 questions right. Bag of words picks out the most similar question if the number of same words are matching. Hence, it may omit the most important keyword of the question and pay more importance to other common words. Therefore, it fails on questions which are longer in size or have too many different words involved. For example, on the third question, it has given equal importance to words like ‘of’ and ‘was’ which made it retrieve the wrong question from the database. Hence, the BOW model is rarely used.

4.2 Word2Vec

Word2Vec embeddings are popularly trained using the skip-gram model. These embeddings are trained to take a word as input and reconstruct its context. As a result, they are able to take into account semantic similarity of words based on context information. The resulting embeddings are such that words with similar meaning tend to be closer in terms of cosine similarity. The interesting property of word2vec is that it is able to approximate meanings of words and it can use vector algebra to obtain the meaning of a question not exactly present in the database from 2 or 3 other similar ones which are present.
For example, vector(’Paris’) - vector(’France’) + vector(’Italy’) results in a vector that is very close to vector(’Rome’).
For hyperparameter tuning, we are specifically using the word2vec-google-news-300 (which is the

most commonly used pre-trained model based on the Google News dataset that has 3 billion running words and creates 300 dimensional embedding for 3 Million words) which is a pre-trained model which creates a word embedding for each question which is then conveniently added for each word to get the phrase embeddings. Then we retrieve the answer to the closest possible embedded question in our dataset using the same model.



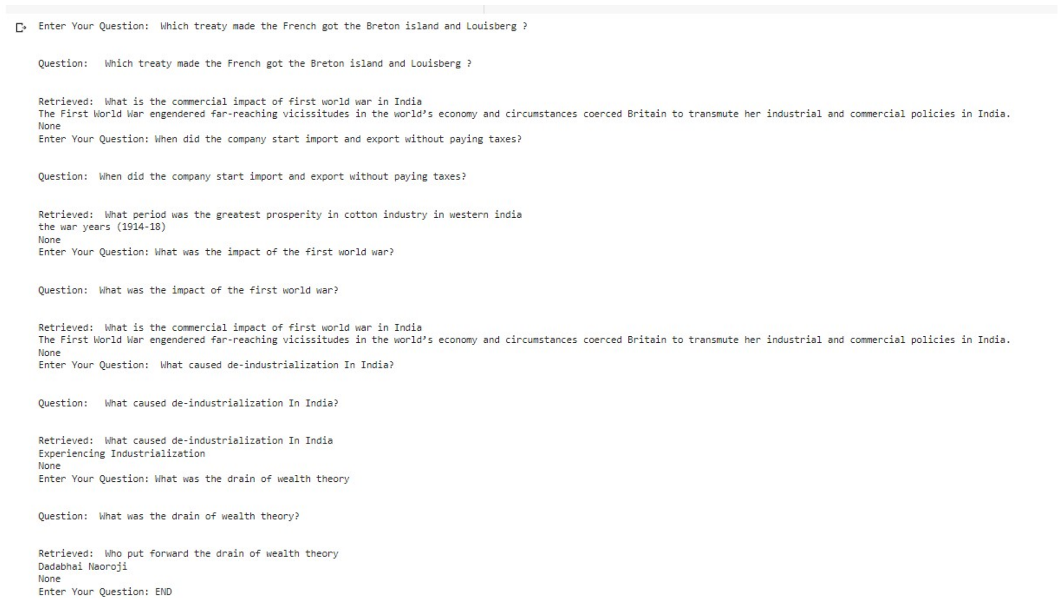
```
        print(answer(question))

Enter Your Question: Which treaty made the French got the Breton island and Louisberg ?


Question:  Which treaty made the French got the Breton island and Louisberg ?


Retrieved:  Which treaty made the French got the Breton island and Louisberg in 1748
Aix la chappelle
None
Enter Your Question: When did the company start import and export without paying taxes?


Question:  When did the company start import and export without paying taxes?


Retrieved:  When did the company get liberty to export and import goods in India without paying taxes
1717
None
Enter Your Question: What was the impact of the first world war?


Question:  What was the impact of the first world war?


Retrieved:  What is the commercial impact of first world war in India
The First World War engendered far-reaching vicissitudes in the world's economy and circumstances coerced Britain to transmute her industrial and commercial policies in India.
None
Enter Your Question: What caused de-industrialization In India?


Question:  What caused de-industrialization In India?


Retrieved:  What caused de-industrialization In India
Experiencing Industrialization
None
Enter Your Question: What was the drain of wealth theory


Question:  What was the drain of wealth theory?


Retrieved:  What was the drain of wealth theory
Impression in India, according to them, was the result of a steady drain of Indian wealth into Britain-a result of British colonial policy. This drain occurred through the interest that India paid for peregrine debts of the East India Company, military expenditure, ensured returns on peregrine investment in railways and other infrastructure,
None
Enter Your Question: END
```

On comparing with actual answers, we can see that it got 5 out of 5 correct answers. This is because of the fact that word2vec is trained on a very large corpora and hence is able to properly understand the semantics of the question, even if it is slightly different from the one in database.

4.3 GloVe

GloVe is an unsupervised learning algorithm developed by Stanford University for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.
GloVe is essentially a log-bilinear model with a weighted least-squares objective. The main intuition underlying the model is the simple observation that ratios of word-word co-occurrence probabilities have the potential for encoding some form of meaning. The training objective of GloVe is to learn word vectors such that their dot product equals the logarithm of the words' probability of co-occurrence. Owing to the fact that the logarithm of a ratio equals the difference of logarithms, this objective associates (the logarithm of) ratios of co-occurrence probabilities with vector differences in the word vector space. Because these ratios can encode some form of meaning, this information gets encoded as vector differences as well. For this reason, the resulting word vectors perform well on word analogy tasks, which is similar to its use in our application above for question-answer retrieval. Here we are using the glove-twitter-25 , which creates a 25 word embedding for each word vector which is individually added to get phrase embedding similar to the previous case. Then we retrieve the answer to the closest possible embedded question in our dataset.

On some datasets, glove performs better whereas on others, word2vec may perform better. Here are the answers to the 5 questions asked in the previous two cases also retrieved using glove.



```
Enter Your Question:  Which treaty made the French got the Breton island and Louisberg ?


Question:   Which treaty made the French got the Breton island and Louisberg ?


Retrieved:  What is the commercial impact of first world war in India
The First World War engendered far-reaching vicissitudes in the world’s economy and circumstances coerced Britain to transmute her industrial and commercial policies in India.
None
Enter Your Question: When did the company start import and export without paying taxes?


Question:  When did the company start import and export without paying taxes?


Retrieved:  What period was the greatest prosperity in cotton industry in western india
the war years (1914-18)
None
Enter Your Question: What was the impact of the first world war?


Question:  What was the impact of the first world war?


Retrieved:  What is the commercial impact of first world war in India
The First World War engendered far-reaching vicissitudes in the world’s economy and circumstances coerced Britain to transmute her industrial and commercial policies in India.
None
Enter Your Question:  What caused de-industrialization In India?


Question:   What caused de-industrialization In India?


Retrieved:  What caused de-industrialization In India
Experiencing Industrialization
None
Enter Your Question: What was the drain of wealth theory


Question:  What was the drain of wealth theory?


Retrieved:  Who put forward the drain of wealth theory
Dadabhai Naoroji
None
Enter Your Question: END
```

We can see that it provided 2 out of 5 correct answers, with the last one being moderately correct. This is owing to the nature of glove and the relatively smaller corpora it was trained upon.

Although in this passage, their performances are similar, Word2vec provides better results than glove in real life applications as the size of the word embeddings are bigger and also it is trained on bigger corpora.

4.4 Conclusion

We can see that glove and word2vec are better than the naive bag-of-words model in rule based question answering systems. Moreover, the accuracy of glove and word2vec depends on the hyperparameters chosen for each. Also for this, the size of the dataset we used was extremely important. In general, the size of the dataset matters the most for rule based approach and the more questions it contains, better the results. Our dataset had 100 questions hence the results are moderate but it is very difficult to construct it for large passages. Moreover our passage is based on history texts with many historical Indian and British names of people, battles and places, many of which wouldn’t have occurred in the corpora used to train word2vec or glove models. Hence for the given passage, their accuracy is lower when compared to actual use cases where these models are used.

Rule based systems are rarely used in real life question answering systems, but they are useful when an application demands only a small number of specific predetermined tasks to be carried out and these questions are easily available in the small FAQ dataset. For example, Jio has enabled mobile recharging through whatsapp, for this, a rule based method can be employed to convert the natural language query into a word2vec or glove vector and a specific task(like recharge or balance check or a small number of other tasks) can be carried out after retrieval of correct mapped answer.

5 Results

Question-answering based on the given passage is implemented successfully using BERT and one traditional method for QA. Both the methods performed well given some circumstances. BERT based QA takes more time and space and traditional method for QA takes less time but it depends on the Question-answer pair database available. Also the current implemented versions are not able to identify out of context questions. For BERT based QA, it can answer accurately most of the time if attention span needed is 1-2 lines but may as the required attention span increases the accuracy may decrease. Overall if we compare both the methods then we find that with sufficiently available question-answer pair database, the traditional method for QA may perform well but this requirement of comprehensive database of QA-pair pose a major limitation on its use in general. This is one of the limitations that motivates the need for research in deep learning methods for NLP. In recent times transformers based methods have been emerged as state of the art methods for many NLP tasks including question-answering.

6 Individual contribution

1. Gautam Sagar: Implemented BERT based QA system.(Section 3 of the report)
2. Razorshi Prozzwal Talukdar: Implemented Traditional QA system.(Section 4 of the report)
3. Yashas Rohan R: Implemented Traditional QA system.(Section 4 of the report)
4. Vaibhav Vishal: Case Study(section 2 of the report)