# Evaluation of different benchmark Models for Question-Answering System using FAQs.

Avineet Kumar Singh

College of Engineering and Computing

University of South Carolina

Email: AS89@email.sc.edu

**BACKGROUND**

In the latest and most important applications of NLP we focus on how we could improve our question-answering system, be it a chatbot which takes input in a text format and gives output in text format or a device like Alexa which takes input in audio format and gives output in audio format. Both the devices require good accuracy to make it useful and worthy.

For creating any application related to search engine or for any customer service-related applications like chatbots/Question Answering Systems, we need to compute the similarity between text. These similarities need to be beyond just the comparisons in overlapped words. Such applications should be able to understand semantically similar queries from users. For example, question-and-answer sites such as Quora or Stackoverflow need to determine whether a question has already been asked before.

So, the Text similarity must determine how 'close' two pieces of text are both in surface closeness (word level similarity) and meaning (semantic similarity).

Now these texts/documents are converted to a vector of features (word embeddings) and then compared by measuring the distance between these vectors.

For converting sentences to Vectors we could use any of the following methods.

1) Bag of Words (BoW)

2) Term Frequency - Inverse Document Frequency (TF-IDF)

3) Continuous BoW (CBOW) model and SkipGram model embedding (SkipGram)

4) Pre-trained word embedding models :

   a) Word2Vec (by Google)

   b) GloVe (by Stanford)

   c) fastText (by Facebook)

5) Poincarré embedding

6) Node2Vec embedding based on Random Walk and Graph

For measuring the distance between Vectors, we could use any of the following methods.

1) Jaccard Similarity

2) K-means

3) Cosine Similarity

4) Jensen-Shannon distance

5) Word Mover Distance

6) Variational Auto Encoder (VAE)

7) Universal sentence encoder

8) Siamese Manhattan LSTM

Lexical word alignment is also a good method for comparing sentence similarity. However, vectors are more efficient to process and allow to benefit from existing ML/DL algorithms.

In this project I have focused on Cosine similarity for comparing different vectors and below are the methods which is used for creating word embeddings.

1) Bag of Words (BoW)

2) Pre-trained word embedding models :

a) Word2Vec- SkipGram model

b) GloVe

3) BERT embeddings

Skip-Gram (aka Word2Vec) and Glove are static word embeddings whereas BERT (Bidirectional Encoder Representations from Transformers) is a Contextualized (Dynamic) Word Embedding.

**Cosine similarity** is one of the most widely used and powerful similarity measures in Data Science. It is used in multiple applications such as finding similar documents in NLP, information retrieval, finding similar sequence to a DNA in bioinformatics, detecting plagiarism and may more. It takes the angle between two non-zero vectors and calculates the cosine of that angle, and this value is known as the similarity between the two vectors. This similarity score ranges from 0 to 1, with 0 being the lowest (the least similar) and 1 being the highest (the most similar).

Cosine similarity is calculated as follows,

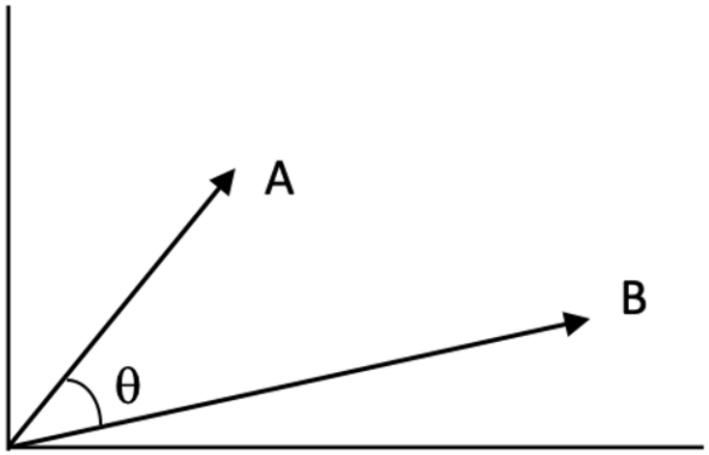


Fig: Angle between two 2-D vectors A and B

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Formula: calculation of cosine of the angle between A and B

Leveraging the **Bag of Words (BOW)** is the first paradigm we can use for semantic similarity. With BOW, each sentence is encoded into a vector whose length in the vocabulary is the number of terms. Each element of the vector shows how many times in the sentence the specific word occurs. For the classification of documents as a whole, BOW or TF-IDF is good.

The first model we will use for semantic similarity is leveraging Bag of Words (BOW). With BOW, each sentence is encoded into a vector whose length is the number of words in the vocabulary. Each element of the vector indicates how many times the particular word occurs in the sentence. BOW or TF-IDF is good for classification of documents as a whole.

Word embeddings are good for identifying contextual content.

**Word2Vec** embeddings are popularly trained using the skipgram model. It is the most common model for Word2vec. In particular, the pre-trained model most widely used is based on the Google News dataset of 3 billion running words and 3 million words of up to 300-dimensional embedding. Such embeddings are equipped to take a word and reconstruct its context as an input. 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.

**Glove** is an alternative approach to constructing word embeddings on the word-word co-occurrence matrix using matrix factorization techniques.

Although both techniques are common, on some datasets, glove performs better, while on some, word2vec skipgram model performs better. Here, both the word2vec and glove models are being experimented with.

**BERT**, a transformer-based paradigm, instead of looking at words in isolation, tries to use the meaning of words to get embedded. In 2018, BERT broke many records in NLP tasks, a major NLP leap. In deep learning, BERT uses many principles to create a model that looks at meaning in a bi-directional way, using knowledge from the entire sentence as a whole through self-attention. The language modeling tools such as ELMO, GPT-2 and BERT allow for obtaining word vectors that morph knowing their place and surroundings.

Pre-training a BERT model is a fairly expensive yet one-time procedure for each language. Fortunately, Google released several pre-trained models. It can be downloaded, and they have scripts to run BERT and get the word vectors from any and all layers. The base case BERT model that we use here employs 12 layers (transformer blocks) and yields word vectors with p = 768.

BERT word vectors morph themselves based on context.

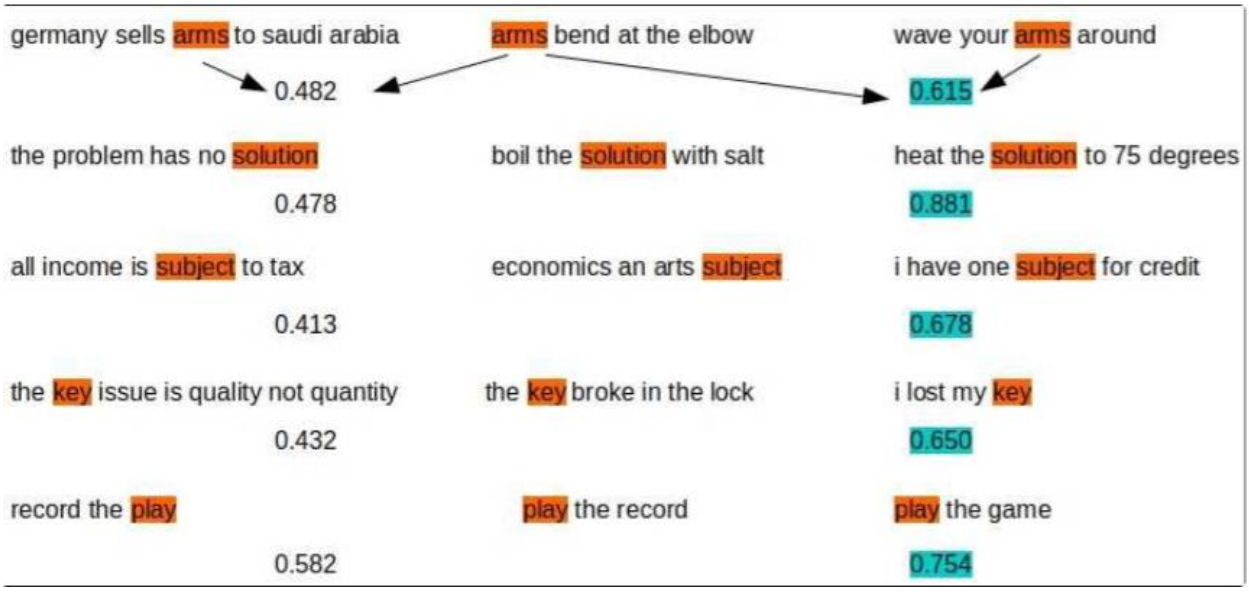


**Diagram:** BERT embeddings are contextual. Each row shows three sentences. The sentence in the middle expresses the same context as the sentence on its right, but different from the one on its left. All three sentences in the row have a word in common. The numbers show the computed cosine-similarity between the indicated word pairs. BERT embedding for the word in the middle is more like the same word on the right than the one on the left.

Below are few examples in the recent developments in Question Answering system for comprehension/document.

SQuAD datasets are some of the famous datasets which are used to analyze such systems.

The best system(model) by the end of 2019-02-07 using **SQuAD v1.1,,** was

1) BERT (ensemble)

Date: Oct 05, 2018

F1: 93.160

2) BERT (single model)

Date: Oct 05, 2018

F1: 91.202

The Human Performance on this had an F1 score of 91.221

Defects of SQuAD 1.0 is that all the questions have an answer in the paragraph.

The human performance was not judged properly, as it collected only three sets of answers.

In SQuAD 2.0, 1/3 of the training questions have no answer, and about ½ of the dev/test questions have no answer.

So, some of SQuAD 1.0 systems were taken to check how well they perform on SQuAD 2.0, thus the systems which did well before did not perform well, like BiDAF system.

The best system(model) by the end of 2019-02-07 using **SQuAD v2.0**,, was

1) BERT + MMFT + ADA (ensemble)

Date: Jan 15, 2019

F1: 87.615

2) BERT + Synthetic Self-Training (ensemble)

Date: Jan 10, 2019

F1: 86.967

The Human Performance on this had an F1 score of 89.452

Limitations of SQuAD dataset

1) It picks up only span based answers (no yes/no, counting, implicit why) and not the reasoning based answers.

2) All questions were based on the passage, both in the terms of the words used and the syntactic structure matching. This makes question answering naturally easy. In real world, the users may just type something.

**PROBLEM**

Often websites have comprehensive FAQs, but manually searching and finding the answer to a specific question from these FAQs is not trivial. Examining main family of models is important in understanding what all things can be done to make the model better. Which model to implement for a smaller chatbot projects which could be more feasible in terms of space instead of loading heavy models.

**APPROACH**

In this project, I examined the task of automatically retrieving a suitable response to customer questions from FAQs. My basic strategy is as follows: For a given query, find the FAQ question that is closest in meaning to the user query and display it to the user. For this, we need to have an efficient way of computing **semantic similarity** between two sentences.

To compute semantic similarity between sentences, we will convert each sentence into a vector. I can then use cosine similarity between vectors to come up with a distance measure between sentences that indicates how similar they are in meaning. In this project, below models were used
1) Bag of Words (BoW)

2) Pre-trained word embedding models :

a) Word2Vec- SkipGram model

b) GloVe

3) BERT embeddings

The basic approach is to use **Different Embeddings + Cosine Similarity** on different datasets.

I have used different types of dataset for the four models. They are as follows:

1) Big dataset with 85282 FAQs.

It was downloaded using following link

https://github.com/LasseRegin/medical-question-answer-data

This dataset will help us to determine how much time a model takes to create word embeddings with large datasets.

2) Easy dataset with 10 FAQs

This was created by converting 10 FAQs using a paraphrasing tool called QuillBot.

3) Medium dataset with 10 FAQs

This was created by manually paraphrasing 10 FAQs keeping 3 to 4 words in common.

4) Hard dataset with 10 FAQs

This was created by manually paraphrasing 10 FAQs keeping 1 to 2 words in common.

The smaller datasets would help us study the accuracy of different models based on the level of paraphrasing.

I have also implemented a chatbot type layout for different models to judge the response time as well for each question asked on larger dataset.

**DEMONSTRATION**

Code is uploaded on GitHub (https://github.com/AVINEET-Singh/NLP_Project)

In Data Preprocessing, I have only kept text in the sentences and removed everything else. Stopwords are not removed as models use contextual context.

**In Bag of Words model,**

Sentences are converted to vectors which look as below. For each word a vector is created where first number is the position of that word in the whole document and second number is the frequency of that word in the sentence.

```
how is the job searching experience nowadays
[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1)]
```

Using Cosine similarity, we determine similarity of input question with each of the available questions(sentences)

```
Question:  how is the job searching experience nowadays
```

```
1.0 what does the job hunting experience look like
0.31622776601683794 any insights you can offer about the ds job market
0.1643989873053573 whats the impact of covid on hiring for ds roles
0.19611613513818404 what skills and qualities do employers look for in a data scientist
0.19611613513818404 do employers look for an advanced ml degree
1.0 how does a typical day of a data scientist look like
0.058722021951470346 is preparation of algorithms and data structures needed for a data science interview
0.1643989873053573 what is the mathematical background required to be a data scientist
0.1643989873053573 what are the various rounds in a data scientist interview
0.14142135623730951 what level of proficiency is needed for a data scientist in coding
```

Since the cosine similarity is highest for first question the below answer is received.

```
Retrieved:  What does the job hunting experience look like ?
```

A small chatbot using Bag of Words implemented on larger dataset looks like below.

```
Welcome to the Question Answering System
Question:
How is your health?
Answer: you get at least 5 years from a booster so you should be ok if you cleaned the wound well. it is not the germs of tetanis but the "eggs"
----------------------
Response Time 18.00569987297058
----------------------


Enter Quit to exit:
Question:
How to eat which keeps you healthy for longer time?
Answer: you get at least 5 years from a booster so you should be ok if you cleaned the wound well. it is not the germs of tetanis but the "eggs"
----------------------
Response Time 19.53119659423828
----------------------


Enter Quit to exit:
Question:
Quit
----------------------
Thank You
----------------------
```

**In Word2Vec-Skipgram model,**

The pretrained model is downloaded, using API.

```
v2w_model = api.load('word2vec-google-news-300')
```

```
[================================================-] 98.6% 1639.0/1662.8MB downloaded
```

Sentences are converted to 300 dimensional vectors which look as below.

```
[array([[ 6.03637695e-02,  2.61230469e-01, -2.76412964e-01,
        -9.80224609e-02, -1.46474838e-02,  2.96997070e-01,
         2.23632812e-01, -4.87899780e-01, -7.87963867e-02,
         3.68164062e-01,  5.26367188e-01,  6.27441406e-02,
        -4.80712891e-01, -4.44824219e-01, -5.27709961e-01,
         2.36656189e-01,  1.01043701e-01,  1.29296875e+00,
         3.66668701e-01, -1.17724609e+00, -1.08642578e-01,
         9.76562500e-03, -1.66564941e-01,  3.57360840e-01,
         6.57653809e-02, -8.37402344e-02, -4.52514648e-01,
         1.65039062e-01,  1.20361328e-01,  8.22753906e-02,
        -2.86026001e-01,  2.80029297e-01, -2.41699219e-02,
         3.44116211e-01,  1.89941406e-01,  9.76562500e-04,
         4.40429688e-01, -1.01043701e-01,  4.71557617e-01,
         5.62805176e-01,  6.16348267e-01, -2.64465332e-01,
         5.75141907e-01, -3.64990234e-02, -3.16894531e-01,
        -6.45019531e-01, -3.91540527e-02,  4.61456299e-01,
         2.24121094e-01,  2.64358521e-02, -9.91210938e-02,
        -2.13936806e-01, -3.19824219e-02, -2.79479980e-01,
         2.31536865e-01, -5.93017578e-01,  1.63452148e-01,
        -2.87109375e-01,  5.51025391e-01, -6.43066406e-01,
        -6.03027344e-01,  3.86352539e-01, -7.29003906e-01,
         9.28192139e-02, -1.84082031e-01,  2.48523712e-01,
        -1.54785156e-01,  1.24511719e-02, -2.29003906e-01,
        -9.21325684e-02, -1.74133301e-01,  3.08105469e-01,
         7.63671875e-01, -1.28204346e-01, -6.01562500e-01,
        -2.92724609e-01,  1.45263672e-02,  3.70605469e-01,
         4.88525391e-01,  9.60937500e-01,  2.97912598e-01,
        -2.97851562e-02,  7.29675293e-01, -2.37243652e-01,
         1.10717773e-01, -2.28515625e-01, -5.15991211e-01,
         3.54598999e-01,  1.15966797e-01, -1.22955322e-01,
         2.73681641e-01,  5.33325195e-01, -6.61132812e-01,
         3.09448242e-02,  6.69921875e-01, -6.40869141e-01,
         4.34082031e-01,  2.28271484e-01,  4.10400391e-01,
        -6.40869141e-01, -6.61376953e-01, -4.43115234e-01,
         3.57055664e-02,  1.03808594e+00,  5.88256836e-01,
        -4.89257812e-01,  2.37182617e-01, -4.34722900e-02,
         3.71093750e-02, -4.59960938e-01,  1.72607422e-01,
```

Using Cosine similarity, we determine similarity of input question with each of the available questions(sentences)

```
Question:  how is the job searching experience nowadays
```

```
0.7562886005675723 job hunting experience look like
0.4942790239435815 insights offer ds job market
0.4715838891069917 whats impact covid hiring ds roles
0.5060037232017843 skills qualities employers look data scientist
0.36755536969496566 employers look advanced ml degree
0.4499975012966809 typical day data scientist look like
0.363023840373828 preparation algorithms data structures needed data science interview
0.3204953808919259 mathematical background required data scientist
0.2415454079216197 rounds data scientist interview
0.3390166641785198 level proficiency needed data scientist coding
```

Since the cosine similarity is highest for first question the below answer is received.

```
Retrieved:  What does the job hunting experience look like ?
```

A small chatbot using Skipgram model implemented on larger dataset looks like below.

```
Welcome to the Question Answering System
Question:
How to remain healthy?
Answer: the best way to lose weight is to eat less while eating a balanced diet. that means fruits vegetables whole grains and lean protein. for
----------------------
Response Time 18.334087133407593
----------------------


Enter Quit to exit:
Question:
How to eat which keeps us healthy for longer time?
Answer: i would recommend a healthy high protein low carb low glycemic diet. make sure you eat adequate protein before trying to exercise. your
----------------------
Response Time 18.08911895751953
----------------------


Enter Quit to exit:
Question:
Quit
----------------------
Thank You
----------------------
```

**In Glove model,**

The pretrained model is downloaded, using API.

```
glove_model = api.load('glove-twitter-25')
```

```
[==================================================] 100.0% 104.8/104.8MB downloaded
Downloaded and saved glove model
```

Sentences are converted to vectors which look as below.

```
array([[ -1.48087002,   0.26741999,  -2.04841001,  -0.34614   ,
          1.02741005,   1.14953601,   5.7687    ,  -4.07319993,
         -0.79621401,   0.49223798,   1.43883102,   0.71372998,
        -18.83019972,  -0.31153001,   0.92608001,  -0.64904001,
          2.53637002,  -0.43148005,   1.00748099,  -0.442673  ,
         -0.67447802,   0.23409998,  -3.02219999,   1.79036999,
          0.575291  ]])
```

Using Cosine similarity, we determine similarity of input question with each of the available questions(sentences)

```
Question:  how is the job searching experience nowadays
```

```
0.9763126327942122 job hunting experience look like
0.8410695159329984 insights offer ds job market
0.8594510357137447 whats impact covid hiring ds roles
0.885332275712749 skills qualities employers look data scientist
0.8570678125695309 employers look advanced ml degree
0.9746562857737997 typical day data scientist look like
0.8273612181869587 preparation algorithms data structures needed data science interview
0.7774598582973017 mathematical background required data scientist
0.8515788842707537 rounds data scientist interview
0.8106024469805322 level proficiency needed data scientist coding
```

Since the cosine similarity is highest for first question the below answer is received.

```
Retrieved:  What does the job hunting experience look like ?
```

A small chatbot using Glove model implemented on larger dataset looks like below.

```
Welcome to the Question Answering System
Question:
How to take care of health?
Answer: hi toocute1 those are normal numbers!
----------------------
Response Time 15.49782395362854
----------------------


Enter Quit to exit:
Question:
Which food is healthy?
Answer: although some feel queasy eating red meat when they have abstained for a long while there are no ill health effects. interestingly the gu
----------------------
Response Time 15.411398887634277
----------------------


Enter Quit to exit:
Question:
Quit
----------------------
Thank You
----------------------
```

**In BERT model,**

The pretrained model is downloaded.

```
#Download a model listed below, then uncompress the zip file into some folder, say /tmp/english_L-12_H-768_A-12/
!wget https://storage.googleapis.com/bert_models/2018_10_18/uncased_L-12_H-768_A-12.zip
```

```
--2020-11-16 02:44:17--  https://storage.googleapis.com/bert_models/2018_10_18/uncased_L-12_H-768_A-12.zip
Resolving storage.googleapis.com (storage.googleapis.com)... 74.125.195.128, 74.125.142.128, 74.125.20.128, ...
Connecting to storage.googleapis.com (storage.googleapis.com)|74.125.195.128|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 407727028 (389M) [application/zip]
Saving to: ‘uncased_L-12_H-768_A-12.zip’

uncased_L-12_H-768_ 100%[===================>] 388.84M   161MB/s    in 2.4s

2020-11-16 02:44:19 (161 MB/s) - ‘uncased_L-12_H-768_A-12.zip’ saved [407727028/407727028]
```

Sentences are converted to vectors which look as below.

```
[array([[-1.44323036e-01, -4.63209212e-01,  1.29837438e-01,
         -1.66514188e-01,  5.29237747e-01, -9.45909396e-02,
          8.27755928e-02,  7.27962852e-01, -5.29639542e-01,
         -4.56469864e-01,  2.38594845e-01,  1.02570228e-01,
          2.06873164e-01, -3.02936137e-02, -4.24653322e-01,
          2.32581988e-01,  8.04083198e-02,  4.87814620e-02,
          1.84518188e-01, -1.49268582e-01, -1.28089190e-01,
          1.06596991e-01, -3.67873847e-01,  3.91043097e-01,
         -9.11013260e-02, -1.47713795e-01,  1.33011892e-01,
         -1.90492690e-01,  1.27119690e-01, -6.71361163e-02,
         -1.03558965e-01, -1.40349641e-01, -5.22467196e-02,
          5.44517040e-02, -4.36913908e-01, -1.54996976e-01,
         -2.24679783e-01, -2.10354269e-01, -4.56502378e-01,
          1.90109108e-02, -5.07494986e-01, -1.76392034e-01,
         -2.04477496e-02,  3.97594690e-01, -4.72285479e-01,
         -8.40542614e-02, -2.65267808e-02, -1.13781966e-01,
          5.44574670e-03, -4.17859644e-01, -4.28560138e-01,
          7.64537156e-02, -2.28451043e-01, -1.31400600e-01,
         -2.40475759e-01,  1.98234752e-01,  3.86545539e-01,
         -1.16890222e-01, -2.02209413e-01, -2.43189201e-01,
         -2.68396467e-01, -2.11539924e-01, -3.69322360e-01,
         -3.49355340e-01,  2.25163534e-01, -1.93068400e-01,
          3.56322169e-01,  3.20889413e-01, -1.96332008e-01,
          1.57198727e-01,  4.88313101e-02, -9.89769638e-01,
          1.03623345e-01,  1.62329495e-01, -1.09702492e+00,
         -2.42866009e-01,  1.52189627e-01,  4.88215119e-01,
          7.06517473e-02, -3.86409849e-01, -2.47924358e-01,
          4.40063834e-01, -1.23191252e-01, -1.38213828e-01,
          2.72234321e-01,  2.07539916e-01,  3.03654396e-03,
          1.63628757e-01,  5.53928018e-02,  2.60968566e-01,
         -9.92407352e-02,  9.02730078e-02, -3.15973997e-01,
          5.41750729e-01,  6.76226139e-01, -8.96671414e-02,
          3.76271248e-01, -1.57898873e-01,  1.43744916e-01,
          2.40905657e-01,  3.55326056e-01, -3.19831401e-01,
          7.64833838e-02,  1.64385200e-01,  3.41100633e-01,
```

Using Cosine similarity, we determine similarity of input question with each of the available questions(sentences)

```
Question:  how is the job searching experience nowadays
```

```
0 0.9063097 what does the job hunting experience look like
1 0.7981879 any insights you can offer about the ds job market
2 0.78888816 whats the impact of covid on hiring for ds roles
3 0.8210438 what skills and qualities do employers look for in a data scientist
4 0.831553 do employers look for an advanced ml degree
5 0.84997344 how does a typical day of a data scientist look like
6 0.80970764 is preparation of algorithms and data structures needed for a data science interview
7 0.84893197 what is the mathematical background required to be a data scientist
8 0.84583074 what are the various rounds in a data scientist interview
9 0.8273965 what level of proficiency is needed for a data scientist in coding
```

Since the cosine similarity is highest for first question the below answer is received.

```
Retrieved:  What does the job hunting experience look like ?
```

A small chatbot using BERT model implemented on larger dataset looks like below.

```
Welcome to the Question Answering System
Question:
How to take care of health?
Retrieved:  how to avoid obesity?
Answer: stay active and eat right. also make sure you get sleep and stay clear of excess stress.
----------------------
Response Time 14.445129871368408
----------------------


Enter Quit to exit:
Question:
Give some tips for healthy lifestyle?
Retrieved:  give advise on weight loss programme?
Answer: weight loss is possible. it takes a regular change in habits. adopting a low glycemic diet is helpful for many. regular activity daily ar
----------------------
Response Time 14.472314357757568
----------------------


Enter Quit to exit:
Question:
Quit
----------------------
Thank You
----------------------
```

**RESULTS and EVALUATION**

Embedding time for big dataset (85282 sentences) using BERT model were around 55 minutes.

```
Question Embedding time for Huge dataset 3334.6015825271606
```

Other models took around 30 secs for embeddings for big dataset.

Below is accuracy calculated using Confusion matrix.

Accuracy table

| Sl. No. | Model | Easy Dataset | Medium Dataset | Hard Dataset |
|---|---|---|---|---|
| 1 | BOW | 0.3 | 0.2 | 0.2 |
| 2 | Word2Vec-Skipgram | 1 | 1 | 0.4 |
| 3 | Glove | 0.3 | 0.2 | 0.2 |
| 4 | BERT | 0.9 | 0.9 | 0.4 |

**DISCUSSION**

In this project we examined different family of models in NLP, using different kinds of dataset. This study was important to understand how these models behave differently based on different levels of datasets. Datasets differ by the number of words which were same and the way it was paraphrased. Easy dataset was paraphrased using a bot (available online) and paraphrase for other two was done manually.
From the accuracy table we could se that the BOW and Glove model had similar scores whereas scores of Word2Vec-Skipgram model and BERT were almost same.
Performance of BOW model was low as expected because it looks for similar words in predicting answers. Word2Vec-Skipgram performed better as compared to Glove model, which could be because of the type(domain) of dataset. BERT model performed as expected, but the accuracy was quite low for hard dataset may be because of manual paraphrasing.
Paraphrasing manually could be a challenge in testing these models as it varies from person to person, we could see the performance was equally low for all four models.
In future we could study other latest models as well, the dataset used for evaluation could be improved by increasing the data size and improving the methods of paraphrasing. We could also implement a dataset which has multiple questions in a single sentence, increasing the challenge for models. Hence, we can conclude that the Word2Vec model could be implemented if the processing power is low as its performance is similar to that of BERT model which requires GPU.