Automatic Question Answering system using Word embedding

ANURAG ANKIT & SOUBHIK MONDAL

Automatic Question Answering system using Word embedding

Project thesis submitted
in partial fulfillment of the requirement for the degree of

Bachelor of Technology

by

Anurag Ankit and Soubhik Mondal
18010214 / 18010208

Under the supervision of
Dr. Ramesh Ch. Mishra & Dr. Himangshu Sarma

Department of Electronics and Communication Engineering
Indian Institute of Information Technology
Senapati, Manipur
May, 2022

Abstract

Machine learning and artificial intelligence have left a major impact on our surroundings and the outside world for the past decade. The use of machine learning and neural networks has found its application from minor to complex structural issues which varies from different domains and some factors. One of them is automation. Automation has made a lot of lives easier, if not difficult. The use of automation has been implemented in several domains such as software to websites and it has also been implemented in electronics such as applications that require feedback systems.

One of the applications in this automation is the Automatic Question and Answering System. The crucial idea in this is to generate answers as accurately as possible from a set of particular questions and a given context. The work has been done previously using three models, each proving itself better than the previous ones and using the standardized datasets i.e., Stanford Question and Answering Dataset

An interface is carried out in PyCharm Integrated Development Environment where a custom dataset is prepared by the user and in the user interface of this software the question is entered where the algorithm matches the question with the nearest meaning and a score is generated, whichever question match fetches the maximum score the corresponding answer is fetched as an output.

Declaration

We declare that this submission represents our idea in our own words and where others' ideas or words have been included, We have adequately cited and referenced the source. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/sources in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from proper permission has not been taken when needed.

Date: Anurag Ankit and Soubhik Mondal
18010214 and 18010208

Acknowledgement

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts. We would like to gratefully acknowledge our gratitude to my project guide Dr. Ramesh Chandra Mishra, Assistant Professor, Department of Electronics and Communication Engineering, Indian Institute of Information Technology Senapati, Manipur, and Dr. Himangshu Sarma, Assistant Professor Department of Computer Science and Engineering Indian Institute of Information Technology Sri City, Chittoor Andhra Pradesh. Their truly scientific intuition has made them an oasis of ideas and passion in science, which exceptionally inspired us and enriches our growth as students. As a guide, his enthusiastic approach inspired us a lot. We also shall remain highly obliged towards our parent's support and love because of which I was successfully able to complete this project. We would like to thank my friends and classmates for encouraging me to complete this project and making it successful. We would also like to extend our thankful gratitude to our director sir Prof.Dr.Krishnan Baskar, Indian Institute of Information Technology Manipur for providing us with an excellent and skilled environment with adequate facilities.

Anurag Ankit and Soubhik Mondal

Certificate

This is to certify that the project report titled “Automatic Question Answering system using Word embedding” submitted by Anurag Ankit (18010214) and Soubhik Mondal (18010208), final year B.Tech. student in the Department of Electronics and Communication Engineering, Indian Institute of Information Technology Senapati, Manipur is a record of an original research work carried out by them under my supervision and guidance. The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma.

Dr. Ramesh Chandra Mishra
Assistant Professor
Dept. of Electronics Comm. Engg.
IIIT Senapati, Manipur

Department of Electronics & Communication Engineering
Indian Institute of Information Technology Manipur

Certificate

This is to certify that the project report entitled "Automatic Question Answering system using a custom dataset" was submitted by Anurag Ankit (18010214) and Soubhik Mondal (18010208), final year B.Tech. student in the Department of Electronics and Communication Engineering, Indian Institute of Information Technology Senapati, Manipur is approved for the degree of Bachelor of Technology in Electronics Communication Engineering Department.

Examiners:

Dr. Nagesh Ch. **Assistant Professor** **Dept. of ECE**	**Dr. Murli Manohar** **Assistant Professor** **Dept. of ECE**
Dr. Gaurav Saxena **Assistant Professor** **Dept. of ECE**	**Dr. Subasit Borah** **Assistant Professor** **Dept. of ECE**

Dr. Ramesh Chandra Mishra
Supervisor
Dept. of ECE
IIIT Senapati, Manipur

Contents

Chapter 1

Introduction

1.1 Introduction

The Automatic Question and Answering system using word embedding will work revolving between two keywords. They are query and context. A query is the one that is defined as the question whereas the context is the paragraph from which the question is taken and the answer is to be searched. Of the several algorithms that have been introduced since the beginning of neural attention, a mechanism is the one that proved to be the best fit for this purpose. This application has been mainly designed in accordance, keeping in mind the needs of a high school teacher who has a tedious job of correcting a bunch of assignments and answering scripts right after exams and weekends. As most of the subjects in any typical middle school are fact-based so we need to group the questions accordingly and perform the operations on both the query and context to find the most accurate and related keywords for the answer. Most of the questions will be fact-based or single answered such as who, where, when, and then it will come to reasoning and logical questioning such as why?

1.2 Literature Survey

Several algorithms have been introduced in the past for this Question and Answering System.[1] The first one was the BiDirectional Attention Flow(BiDAF) mechanism. This model has been exclusively based on mapping keywords at several levels and the mapping is performed using pre-determined dictionaries. One of them is the GloVE dictionary. When this model was trained for factoid questions using the

Stanford Question and Answering dataset and then when it was tested out on the NCERT prepared dataset it proved to be quite accurate. It achieved a score of 73.3 in Exact Match and an F1 score of 81.1 on the Stanford Question and Answer dataset.

The next that was introduced is the Embedding from the Language model(ELMo). Similar to a BIDAF model[1] this is also based on embedding words in vector representation. Character level tokens are taken as inputs to bidirectional Long short-term memory(LSTM) which provides word-level embedding.

The third model that was introduced was the BERT. This is an advanced level embedding system where the word representation takes place per the neighboring words or words around them.

1.3 Motivation

The work has been done keeping in mind the needs and requirements of the education sector and witnessing the post circumstances of Covid-19 where most of the teaching and interaction between the students and teachers are taking place in the online mode, a full-fledged interface where an algorithm can be intertwined with a user interface would prove to be very handy and get the job easily done. This can be also used by parents who are home-schooling their kids in the first place. This helps both the parents and teachers do their work while they are busy with their busy schedules and correct the answers at a faster pace. This kind of algorithm-based software system can be used to hold frequent assessments and evaluations can be done to keep a clear check on the overall performance and evaluation of an ongoing school scholar.

1.4 Problem formulation

Although this algorithm has proven itself against the well-designed standardized datasets, and also in the most efficient way with a good E.M. (Exact match) and F1 score there are still a few things to look on into in this work.

- Suppose, according to our Indian standardization we can pick a textbook from any 6th-7th standard textbook and then test the algorithm so that there are better chances of using this in real life.
- Supposedly, the answer to be drawn out of the context has failed to do so. A device has to be set up in that scenario.
- We will focus on the logical and reasoning-based questions rather than the objective-type ones. The previous algorithms have proven themselves against it.

1.5 Contribution

We have used GloVE embeddings and Bag of Words to make a fully automated question answering system where the user will be fetched the most appropriate and closest answer among all the question-answer pairs in the dataset.
We have also made a sample dataset that is in a comma-separated value format(CSV) and is easy to modify the question and answer. Since it is in a CSV format the user will be able to add or remove the question and the corresponding answer hassle-free.

1.6 Organisation of the report

The report is organized into the following parts.
Chapter 2 Fundamentals In this chapter we will discuss the basic terminologies related to initiating the project. Chapter 3 In this chapter we have discussed the implementation of the project and the working theory behind it. Chapter 4 In this chapter we have discussed certain test cases of the project where we have tried out the sample questions with their accuracy to fetch the correct output.Chapter 5 In this chapter we will discuss the future works in this project and increasing its productivity to make it more efficient followed by references in Chapter 6.

Chapter 2

Conceptual Foundations

Previously the models in this domain have been designed using several advanced machine learning algorithms certain. Several advanced algorithm blocks have been used which are quite the best algorithms in the field of machine learning. They are Long Short Term Memory(LSTM), Convolutional Neural Network(CNN), Recurrent Neural Network(RNN), and many others. The datasets that have been prepared before that also have a large variety and domain of questions that have certain real-life modern-day applications like search engines on the web and AI assistants.
We will be using a single embedding block focus to match the custom dataset and use a variety of Python libraries that will help our project to execute in the first place.

2.1 Terminologies related to Question Answering system

In this section, we are going to characterize our dataset. Because we know that there is a variety of questions that can be differentiated into various categories. We will be discussing the nature of the question dataset of the project we are considering. Below are some of the given features on which the question dataset will vary from one to another.

2.1.1 Open vs Closed domain

In open domain lots of questions/queries are gathered into a single container. Therefore, it becomes easy to search for a specific query. In a closed domain, one need not necessarily search for a particular query but an algorithm will search for the answer for that query from a pool of information.

2.1.2 Abstractive vs Extractive

Abstractive features down the exact query and the solution obtained is right to the exact keywords. Extractive features down roughly a part of the solution which is approximately near to the exact match. It is less precise than abstractive.

2.1.3 Factoid vs Non Factoid

Factoid are single answered questions. For e.g.:Questions demanding a date, place, or a time. Non-Factoid is reasoning-based questions. E.g.: Questions demanding a reason be it logical or any particular event.

2.2 Summary

Here we have learned about the various terms and we came to know about the domain of questions that we are going to deal with in the Automatic Question and Answering System. It will be the closed domain, abstractive, and non-factoid questions.

Chapter 3

Proposed System

3.1 Introduction

In this chapter, we are going to discuss the architecture and working of our proposed system which consists of a word embedding layer, GloVE Embeddings, Word2Word Vector, and their working principles. We will also discuss their roles in our project and discuss text preprocessing and dataset along with their operations.

3.2 Architecture

Initially, a dataset is prepared to drop all the possible queries from a user. This dataset will now act as a pool of information from where the comparison will be taking place. Now a user interface will open where the user will enter his query. The query is being sent to the localHost server. The entered query is first made to go through the Bag of Words algorithm where the words in the user query and the queries in the dataset are compared. If any keyword is matched then semantic similarity increases for that question-answer pair.
Then the GloVE dictionary Twitter GloVE 25 comes to play. For a better understanding of the semantic similarity, we use this dictionary. This dictionary contains the words with the same meaning and this meaning carries similar weight. So we can match the queries more accurately and efficiently.

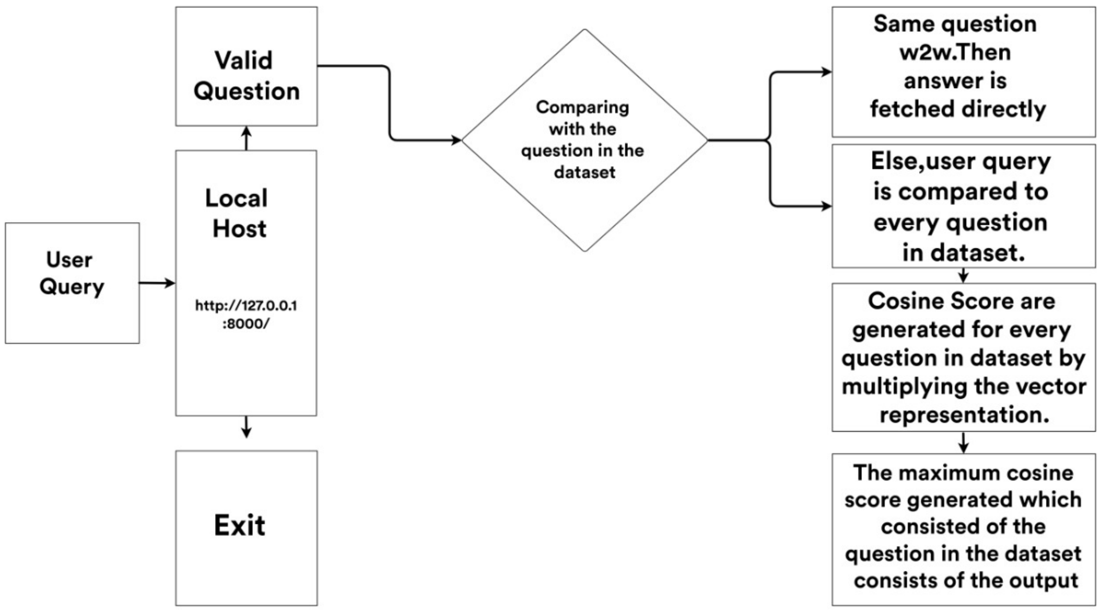

Valid Question
Local Host
http://127.0.0.1:8000/
User Query
Exit
Comparing with the question in the dataset
Same question w2w.Then answer is fetched directly
Else,user query is compared to every question in dataset.
Cosine Score are generated for every question in dataset by multiplying the vector representation.
The maximum cosine score generated which consisted of the question in the dataset consists of the output

Figure 3.1: Architecture Block Diagram of the working principle

3.2.1 Word Embedded Layer

The words/vectors that are identified in the character embedded layer are mapped to a vector space using a pre-trained word embedding model. They are pre-trained GloVEs embeddings to get the vector representations of words in the query and context. From the list of training set there is a word i.e., not present in the list also known as OOV (out of vocabulary) word. GloVE deals with OOV by randomly assigning some values.
The below figure shows the various objects which can be distinguished into various categories and in these categories, certain numeric values are assigned thereby used by the dictionaries to calculate the cosine values.

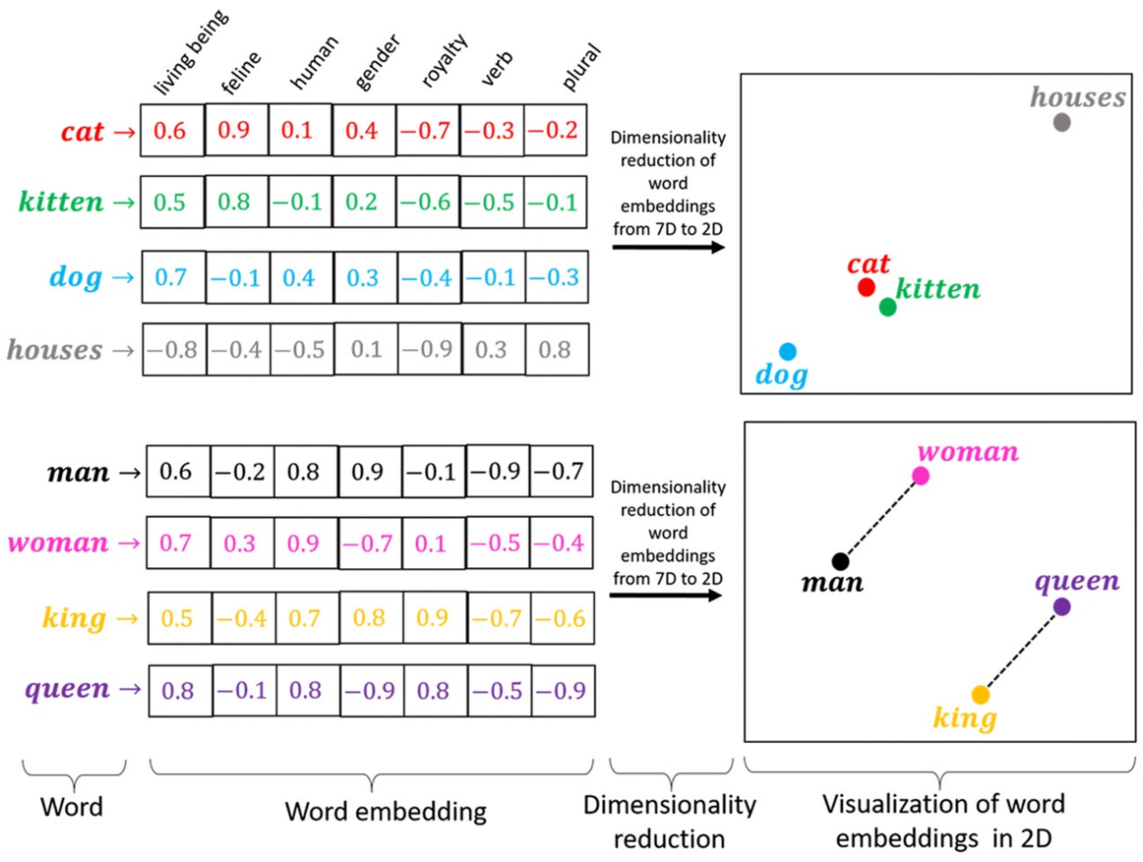

living being
feline
human
gender
royalty
verb
plural
cat → 0.6 0.9 0.1 0.4 −0.7 −0.3 −0.2
kitten → 0.5 0.8 −0.1 0.2 −0.6 −0.5 −0.1
dog → 0.7 −0.1 0.4 0.3 −0.4 −0.1 −0.3
houses → −0.8 −0.4 −0.5 0.1 −0.9 0.3 0.8
man → 0.6 −0.2 0.8 0.9 −0.1 −0.9 −0.7
woman → 0.7 0.3 0.9 −0.7 0.1 −0.5 −0.4
king → 0.5 −0.4 0.7 0.8 0.9 −0.7 −0.6
queen → 0.8 −0.1 0.8 −0.9 0.8 −0.5 −0.9
Dimensionality reduction of word embeddings from 7D to 2D
houses
cat
kitten
dog
woman
man
queen
king
Word
Word embedding
Dimensionality reduction
Visualization of word embeddings in 2D

Figure 3.2: Word embedding layer

3.2.2 GloVE Embeddings

Creating representations of words is to capture their meaning, semantic relationship, and context of different words; here, different word embedding techniques play a role. A word embedding is an approach used to provide dense vector representation of words that capture some context words about their own.
GloVe stands for Global Vectors for word representation. It is an unsupervised learning algorithm developed by researchers at Stanford University aiming to generate word embeddings by aggregating global word co-occurrence matrices from a given corpus.
The basic idea behind the GloVE word embedding is to derive the relationship between the words from statistics. Unlike the occurrence matrix, the co-occurrence matrix tells you how often a particular word pair occurs together. Each value in the co-occurrence matrix represents a pair of words occurring together.
From the figure, the dictionary that we have used is Twitter Glove 25. Here, we have used this dictionary because it has been declared the

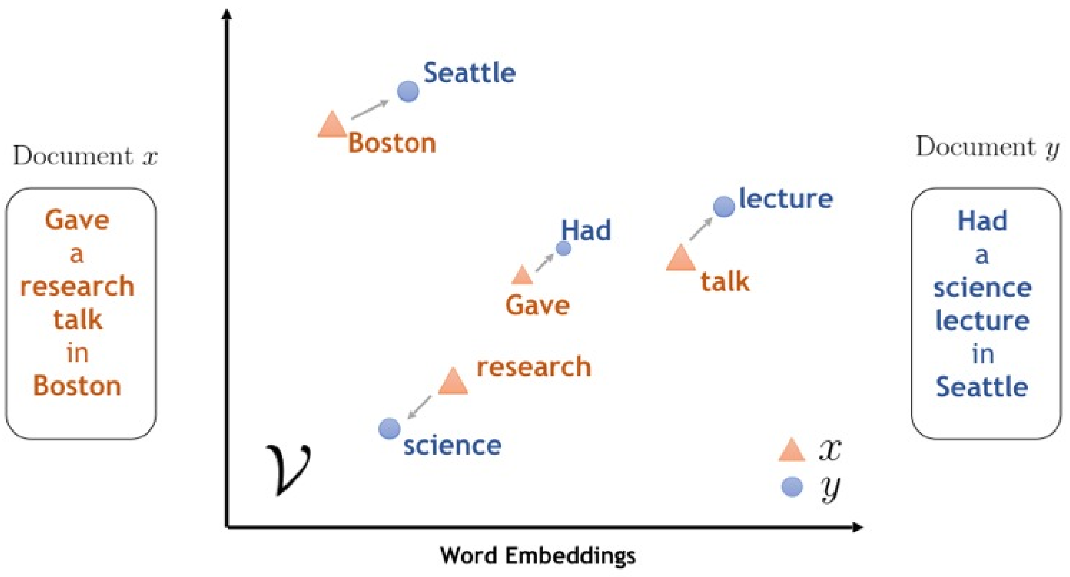


Figure 3.3: GloVE Dictionary

most efficient for this purpose because of its large pool of words.

3.2.3 Word2Vec

Word2vec is a method to efficiently create word embeddings by using a two-layer neural network. It was developed by Tomas Mikolov, et al. at Google in 2013 as a response to make the neural-network-based training of the embedding more efficient and since then has become the de facto standard for developing pre-trained word embedding. The input of word2vec is a text corpus and its output is a set of vectors known as feature vectors that represent words in that corpus. While Word2vec is not a deep neural network, it turns text into a numerical form that deep neural networks can understand.

The Word2Vec objective function causes the words that have a similar context to have similar embedding. Thus in this vector space, these words are close. Mathematically, the cosine of the angle (Q) between such vectors should be close to 1, i.e. angle close to 0. In practice, we use both GloVe and Word2Vec to convert our text into embeddings, and both exhibit comparable performances. Although in real applications we train our model over Wikipedia text with a window size of around 5- 10. The number of words in the corpus is around 13 million, hence it takes a huge amount of time and resources

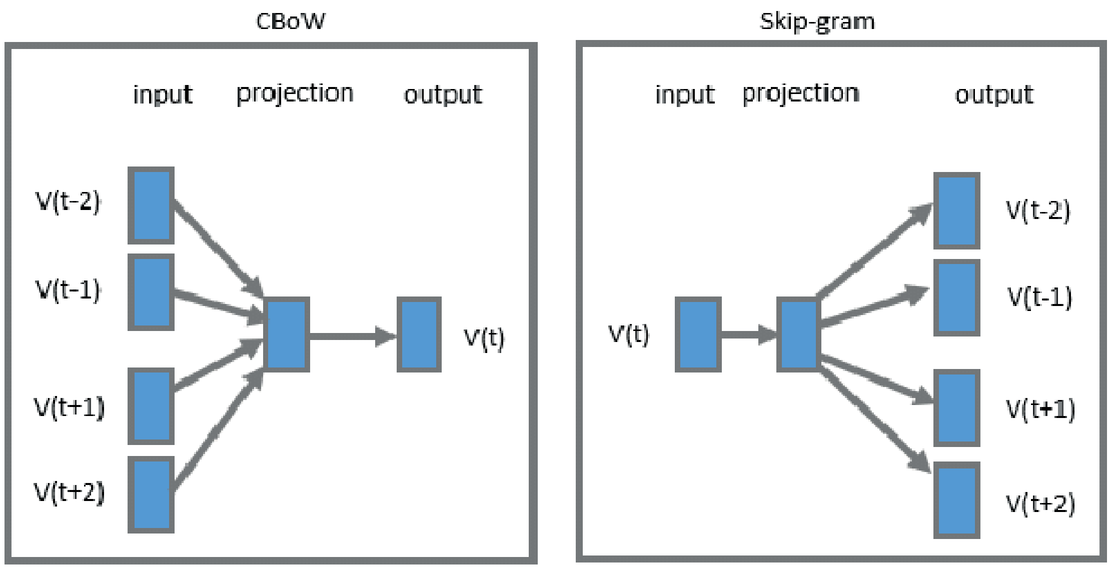


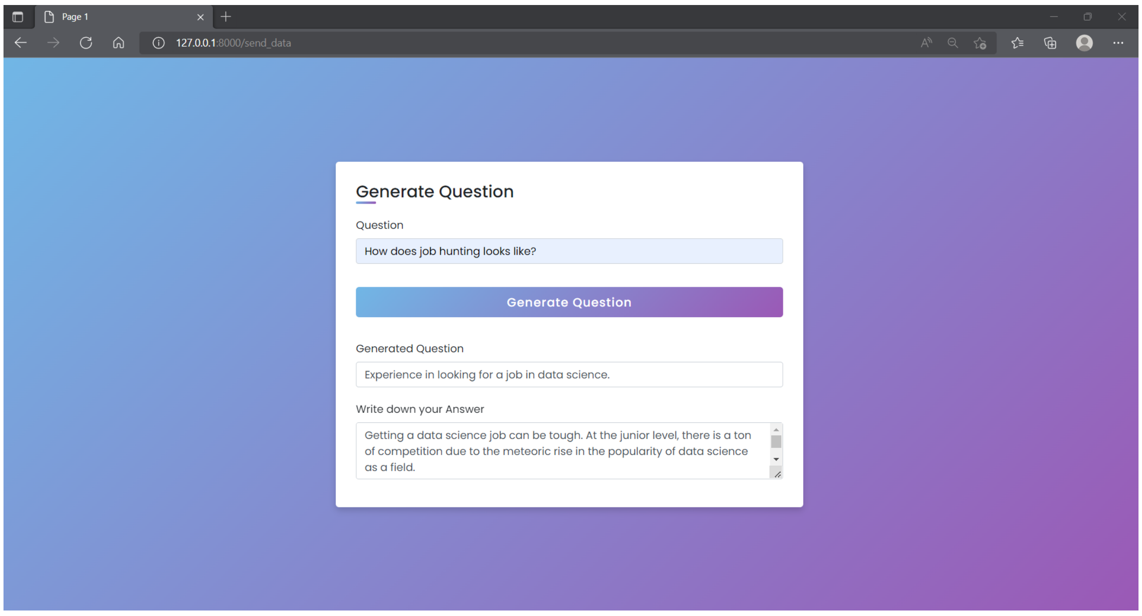
Figure 3.4: Word2Vec

to generate these embeddings. To avoid this we can use the pre-trained word vectors that are already trained and we can easily use them. Here are the links to download pre-trained Word2Vec or GloVe.

3.3 Working

Our 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. We can then use cosine similarity between vectors to come up with a distance measure between sentences that indicates how similar they are in meaning.
Spacy is a well-known library in NLP. It provides functionality for POS-tagging, lemmatization, stop-words, and also embeddings.

The above image shows us the user interface where we enter the user query as we can see there is a column present on the web page where we have entered the query of the user and now we can press the ”Generate Question” to fetch the output.



Question: How does job hunting looks like?

Generated Question: Experience in looking for a job in data science.

Write down your Answer: Getting a data science job can be tough. At the junior level, there is a ton of competition due to the meteoric rise in the popularity of data science as a field.

Figure 3.5: Word2Vec

3.4 Text Preprocessing and Dataset

This is the part where the text data has to summon through some tasks to begin the matching. This is an optional training procedure but performing these operations will bring less time complexity and more efficiency.

3.4.1 Acquire the Dataset

The first step towards our implementation was to decide the domain of our dataset. Initially, when we started our project, we were quite confused about the domain of questions on which we were supposed to work on. Because different kinds of questions are impossible for a single software to face and deploy answers for. Hence, we have targeted the questions in our dataset which were explanatory and logical with explanations that are centered around some topic.
Therefore, our dataset consisted of two independent variables. The QUESTIONS and ANSWERS, the variable QUESTION consists of all the possible questions that can be framed out of a certain topic and the ANSWER consists of the solution to that specific question.

3.4.2 Importing libraries from Python

There are various inbuilt libraries in Python that can perform certain text preprocessing jobs.

- NumPy – It is the fundamental package for scientific calculation in Python. We can use it to add large multidimensional arrays and matrices in code.
- Pandas – It is used for data manipulation and analysis and also for modifying and managing the datasets.

Import the dataset

We have to take the dataset and put it under the working directory. Once we have set the working directory containing the relevant dataset, you can import the dataset using the “read CSV()” function of the Pandas library. This function can read a CSV file (either locally or through a URL) and also perform various operations on it.

dataset= pd.readcsv(‘Dataset.csv’)

Data cleaning

Since, the brain of the computer cannot distinguish words on their own so we have to use our data in such a way that we can get the maximum efficient output out of it, this means we can filter the data in many different ways such that we can fetch only the data that we are supposedly required to process in our algorithm. There are several data cleaning techniques that we can use to do the same. The following are the ones that we have used in our small dataset.

- Lowercasing - The function lowercase converts all the data to lower case because if a question entered is compared to a question in a dataset it can become case-sensitive and cause a conflict hence it can lead to an error that is the wrong solution in the process.
- Removing punctuation – This is used to remove all kinds of symbols because it can cause a hindrance while comparing and categorizing the data.

- Stop words removal – These are words that do not contribute to any meaning in the sentences, so we can remove these words to bring down the overall size of the sentences hence categorizing and comparing becomes easy for the algorithm to take place.

3.4.3 Importance of data preprocessing

The aim of preprocessing the data is to make the dataset more useful so that an algorithm's maximum efficiency can be checked out of it. Hence, it leads to fewer errors, fewer redundancies, fewer missing values, and fewer inconsistencies.

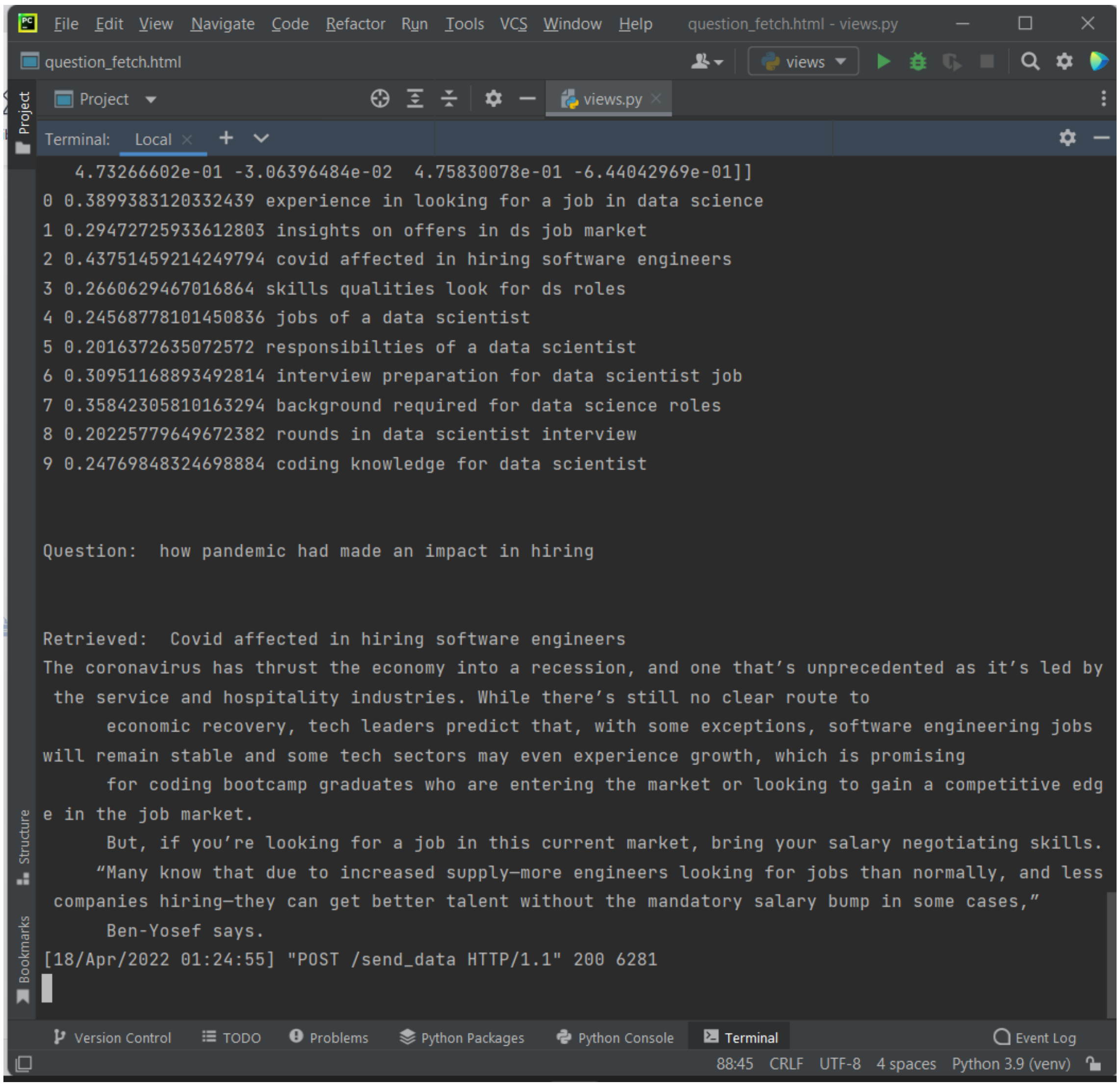
3.5 Results and Discussion

The cosine values between the questions present in the dataset and the query entered by the user is calculated. Whichever question from the dataset has the maximum cosine value concerning the input query the respective answer of the question is fetched as an output. It may not always fetch the correct output but the GloVE twitter 25 dictionary is an efficient one. The words contained in the query and question if similar are given a score of similarity which is a cosine score and these similar words are marked and treated at par with the same manner as much as possible concerning their meanings to find out the most accurate answer to a specific query.

Results

PyCharm is an integrated development environment used in computer programming, specifically for the Python programming language.
Stable Release: Python 2.7, or Python 3.5 or newer It may happen that the test cases will not always generate the correct output so it happened in this case as well. While trying out the variation in question I have encountered two such cases where I fetched a wrong output but the difference in cosine similarities was very less between the wrong and correct output.
Here, the question asks about the rounds of interviews in data science but the answer is fetched related to interview preparation. The correct output is supposed to be question 8 with a cosine score which is

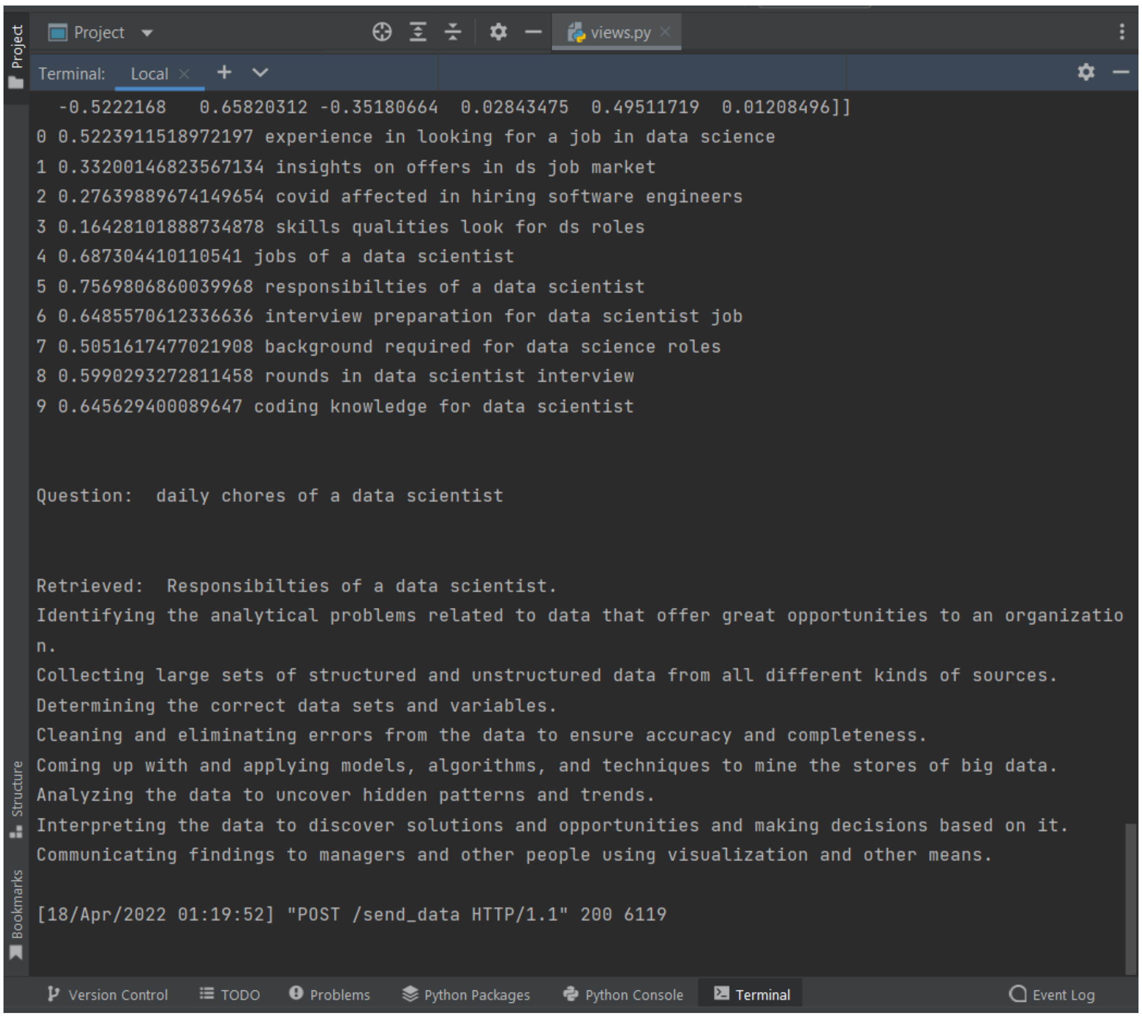
slightly less than the wrong output i.e. question 6. As we can see the right answer has the second maximum cosine score which is not much of a difference but it can be further improvised by re-engineering the GloVE dictionary we have used above.


File Edit View Navigate Code Refactor Run Tools VCS Window Help question_fetch.html - views.py
question_fetch.html
views
Project
views.py
Terminal: Local
Version Control
TODO
Problems
Python Packages
Python Console
Terminal
Event Log
88:45 CRLF UTF-8 4 spaces Python 3.9 (venv)

```
   4.73266602e-01 -3.06396484e-02  4.75830078e-01 -6.44042969e-01]]
0 0.3899383120332439 experience in looking for a job in data science
1 0.29472725933612803 insights on offers in ds job market
2 0.43751459214249794 covid affected in hiring software engineers
3 0.2660629467016864 skills qualities look for ds roles
4 0.24568778101450836 jobs of a data scientist
5 0.2016372635072572 responsibilties of a data scientist
6 0.30951168893492814 interview preparation for data scientist job
7 0.35842305810163294 background required for data science roles
8 0.20225779649672382 rounds in data scientist interview
9 0.24769848324698884 coding knowledge for data scientist


Question:  how pandemic had made an impact in hiring


Retrieved:  Covid affected in hiring software engineers
The coronavirus has thrust the economy into a recession, and one that’s unprecedented as it’s led by
 the service and hospitality industries. While there’s still no clear route to
      economic recovery, tech leaders predict that, with some exceptions, software engineering jobs
will remain stable and some tech sectors may even experience growth, which is promising
      for coding bootcamp graduates who are entering the market or looking to gain a competitive edg
e in the job market.
      But, if you’re looking for a job in this current market, bring your salary negotiating skills.
     “Many know that due to increased supply—more engineers looking for jobs than normally, and less
 companies hiring—they can get better talent without the mandatory salary bump in some cases,”
      Ben-Yosef says.
[18/Apr/2022 01:24:55] "POST /send_data HTTP/1.1" 200 6281
```

Figure 3.6: Case 1

In this figure, we can see that the user input question is ”how the pandemic had made an impact on hiring”. The first algorithm converts user input and dataset questions into vector form. Python library performs the task. After that cosine values are generated between the vector form of the question entered by the user and the question present in the dataset. This is repeated for all questions present in the dataset. We can see here cosine values of all the questions were generated. Maximum cosine value goes to the question ”covid affected in hiring software engineers” is 0.43751459214249794.

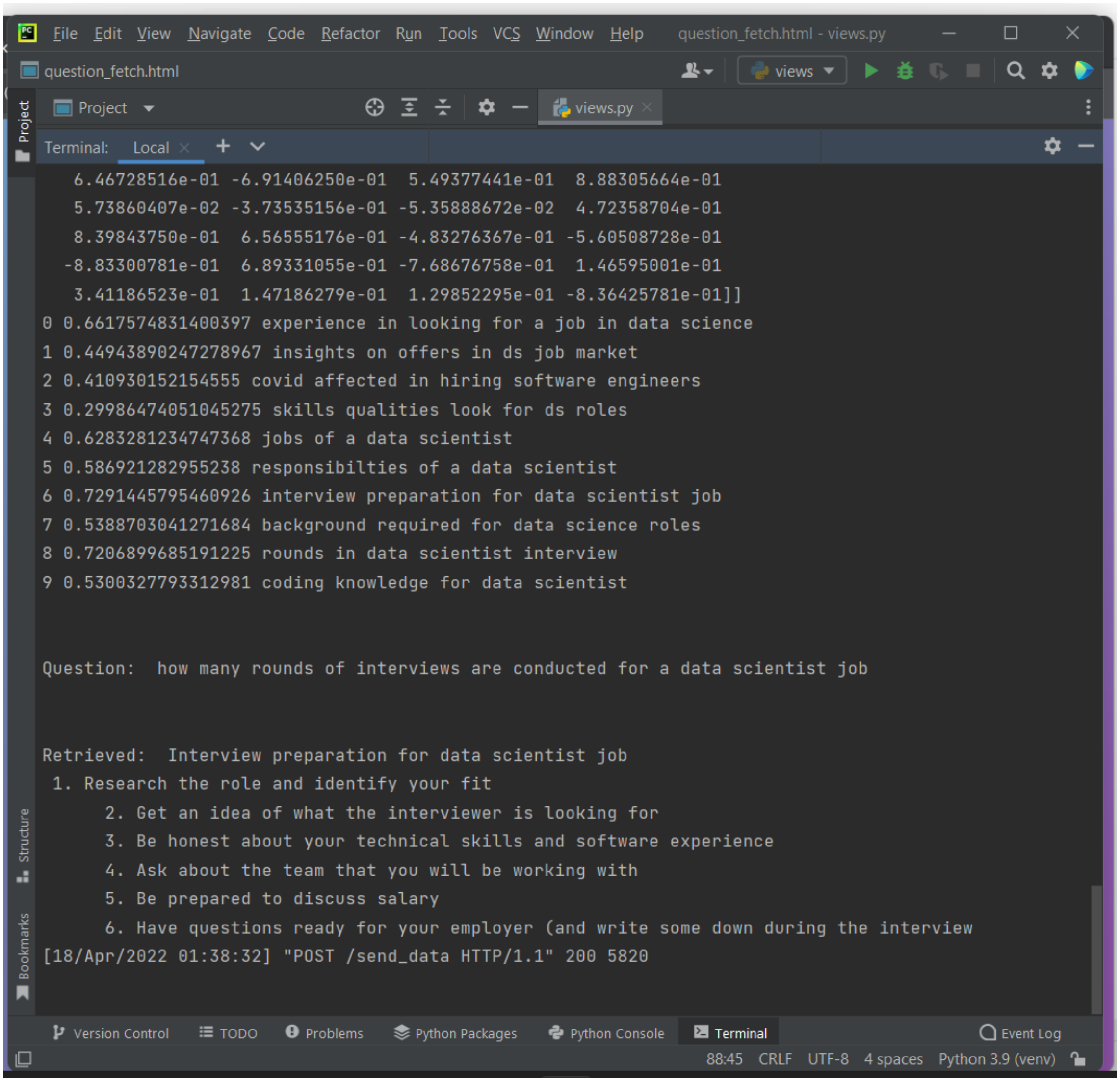


```
  -0.5222168   0.65820312 -0.35180664  0.02843475  0.49511719  0.01208496]]
0 0.5223911518972197 experience in looking for a job in data science
1 0.33200146823567134 insights on offers in ds job market
2 0.27639889674149654 covid affected in hiring software engineers
3 0.16428101888734878 skills qualities look for ds roles
4 0.687304410110541 jobs of a data scientist
5 0.7569806860039968 responsibilties of a data scientist
6 0.6485570612336636 interview preparation for data scientist job
7 0.5051617477021908 background required for data science roles
8 0.5990293272811458 rounds in data scientist interview
9 0.645629400089647 coding knowledge for data scientist


Question:  daily chores of a data scientist


Retrieved:  Responsibilties of a data scientist.
Identifying the analytical problems related to data that offer great opportunities to an organizatio
n.
Collecting large sets of structured and unstructured data from all different kinds of sources.
Determining the correct data sets and variables.
Cleaning and eliminating errors from the data to ensure accuracy and completeness.
Coming up with and applying models, algorithms, and techniques to mine the stores of big data.
Analyzing the data to uncover hidden patterns and trends.
Interpreting the data to discover solutions and opportunities and making decisions based on it.
Communicating findings to managers and other people using visualization and other means.

[18/Apr/2022 01:19:52] "POST /send_data HTTP/1.1" 200 6119
```

Figure 3.7: Case 2

In this figure, we can see that the user input question is ”daily chores of a data scientist”. The first algorithm converts user input and dataset questions into vector form. Python library performs the task. After that cosine values are generated between the vector form of the question entered by the user and the question present in the dataset. This is repeated for all questions present in the dataset. We can see here cosine values of all the questions were generated. The maximum cosine value that goes to the question ”Interview preparation for data scientist job” is 0.7291445795460926.
But the output fetched in this result is the wrong output. We al-

ready have a question in the dataset where the rounds of interviews are asked. Hence, the correct output is not fetched. But let us see how much difference it is making in the correct output. As we can see the second most cosine score is 0.7206899685191225 which is next to the fetched output and this second most is the exact output and the correct answer whose corresponding question is "rounds in data science interview" hence we can say word-level embedding is not the best solution for this but if we can use Long Short Term Memory we can bring out the more accurate and precise results.


File Edit View Navigate Code Refactor Run Tools VCS Window Help question_fetch.html - views.py
question_fetch.html
views
Project
views.py
Terminal: Local
Version Control
TODO
Problems
Python Packages
Python Console
Terminal
Event Log
88:45 CRLF UTF-8 4 spaces Python 3.9 (venv)

```
   6.46728516e-01 -6.91406250e-01  5.49377441e-01  8.88305664e-01
   5.73860407e-02 -3.73535156e-01 -5.35888672e-02  4.72358704e-01
   8.39843750e-01  6.56555176e-01 -4.83276367e-01 -5.60508728e-01
  -8.83300781e-01  6.89331055e-01 -7.68676758e-01  1.46595001e-01
   3.41186523e-01  1.47186279e-01  1.29852295e-01 -8.36425781e-01]]
0 0.6617574831400397 experience in looking for a job in data science
1 0.44943890247278967 insights on offers in ds job market
2 0.410930152154555 covid affected in hiring software engineers
3 0.29986474051045275 skills qualities look for ds roles
4 0.6283281234747368 jobs of a data scientist
5 0.586921282955238 responsibilties of a data scientist
6 0.7291445795460926 interview preparation for data scientist job
7 0.5388703041271684 background required for data science roles
8 0.7206899685191225 rounds in data scientist interview
9 0.5300327793312981 coding knowledge for data scientist


Question:  how many rounds of interviews are conducted for a data scientist job


Retrieved:  Interview preparation for data scientist job
 1. Research the role and identify your fit
      2. Get an idea of what the interviewer is looking for
      3. Be honest about your technical skills and software experience
      4. Ask about the team that you will be working with
      5. Be prepared to discuss salary
      6. Have questions ready for your employer (and write some down during the interview
[18/Apr/2022 01:38:32] "POST /send_data HTTP/1.1" 200 5820
```

Figure 3.8: Failure Case

In this figure, we can see that the user input question is "how many rounds of interviews are conducted for a data scientist job". The first algorithm converts user input and dataset questions into vector form. Python library performs the task. After that cosine values are generated between the vector form of the question entered by the user and the question present in the dataset. This is repeated for all questions present in the dataset. We can see here cosine values of all the questions were generated. The maximum cosine value goes to the question "Interview preparation for a data scientist job " is 0.43751459214249794.

3.6 Summary

In the above cases, we have seen that the implementation almost works for every set of input and also acts as a checker. The algorithm can almost face every kind of output it is being thrown at and we have also seen cases where we have found error outputs with very less difference in the results. Slowly we can improve this by adding more dictionaries and Machine learning prediction algorithms that are used in unsupervised learning.

Chapter 4

Future work and Conclusion

The first step for future work on our model is to identify the bugs in our code that are causing the model to run so slowly and to be unable to accommodate a batch size greater than four. The next step is to identify the bugs that are causing our model to obtain such low scores. We believe that our model design is reasonable and are confident that there is some issue in our code, likely our use of the TensorFlow API, that is causing poor results, rather than our conceptual understanding. Our code is structured into a core QA model class, and all of our models inherit from this class.
The day before this paper was submitted, we believed that the issue was in our QA model superclass or its support code. This was because all the models we had tried were failing. But now that we finally have a reasonably working baseline (which also subclasses QA Model) we believe it is more likely that the problem lies in our specific model implementations.
Now when we have finished implementing the model with word embeddings we can focus more on our structure and use core machine learning algorithms to predict the answer to a certain question like Long Short Term Memory where we can use memory-based data allocation for saving previous data. These question and answer previously-stored has also been used in digital assistant web browsers. So this algorithm will fit well into our system to make it more efficient and larger.

Chapter 5

Reference

[1] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, Hannaneh Hajishirzi. 2017. Bidirectional Attention Flow for Machine Comprehension. DOI: https://arxiv.org/abs/1611.01603

[2] Antoine Bordes, Sumit Chopra, Jason Weston. Question Answering with Subgraph Embeddings. DOI: https://arxiv.org/abs/1406.3676

[3]Question Answering with a fine-tuned BERT. DOI: https://towardsdatascience.com/question-answering-with-a-fine-tuned-bert-bc4dafd45626

[4]Chris Lemke.Data preprocessing in NLP. https://towardsdatascience.com/data-preprocessing-in-nlp-c371d53ba3e0

[5]NLP Tutorials Part -I from Basics to Advance. Abhishek Jaiswal — January 13, 2022. https://www.analyticsvidhya.com/blog/2022/01/nlp-tutorials-part-i-from-basics-to-advance/

[6] Hands-On Guide To Word Embeddings Using GloVe.https://analyticsindiamag.com/hands-on-guide-to-word-embeddings-using-glove/, PUBLISHED ON AUGUST 17, 2021 Hands-On Guide To Word Embeddings Using GloVe.