A PROJECT REPORT ON

“Healthcare Chatbot”

SUBMITTED TO

SHIVAJI UNIVERSITY, KOLHAPUR

IN THE PARTIAL FULFILLMENT OF REQUIREMENT FOR THE AWARD OF DEGREE BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

MR. JUBER ABDUL MOMIN	18UCS067
MR. SUSHANT SURESH PATEKARI	18UCS071
MR. PRAJWAL NANDKUMAR PATIL	18UCS079
MR. SHIVSHANKAR YASHWANT PATIL	18UCS081
MR. SUMERU SHITALKUMAR PATIL	18UCS083

UNDER THE GUIDANCE OF

Prof. K. S. KADAM

DKTE

Promoting Excellence in
Teaching, Learning & Research

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DKTE SOCIETY’S TEXTILE AND ENGINEERING INSTITUTE,

ICHALKARANJI

2021-2022

D.K.T.E.SOCIETY'S

TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI
(AN AUTONOMOUS INSTITUTE)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DKTE
Promoting Excellence in
Teaching, Learning & Research

`

CERTIFICATE

This is to certify that, project work entitled

"Healthcare Chatbot''

is a bonafide record of project work carried out in this college by

MR.	**JUBER ABDUL MOMIN**	**18UCS067**
MR.	**SUSHANT SURESH PATEKARI**	**18UCS071**
MR.	**PRAJWAL NANDKUMAR PATIL**	**18UCS079**
MR.	**SHIVSHANKAR YASHWANT PATIL**	**18UCS081**
MR.	**SUMERU SHITALKUMAR PATIL**	**18UCS083**

is in the partial fulfillment of the award of degree Bachelor in Technology in Computer Science & Engineering prescribed by Shivaji University, Kolhapur for the academic year 2021-2022.

Prof. K.S.Kadam
(PROJECT GUIDE)

PROF.(DR.) D.V.KODAVADE
(HOD CSE DEPT.)

PROF.(DR.) P.V.KADOLE
(DIRECTOR)

EXAMINER: _________________

`

DECLARATION

We hereby declare that, the project work report entitled “**Healthcare Chatbot**” which is being submitted to D.K.T.E. Society’s Textile and Engineering Institute Ichalkaranji, affiliated to Shivaji University,Kolhapur is in partial fulfillment of degree B.TECH. It is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for the award of any degree. Further, we declare that we have not violated any of the provisions under the Copyright and Piracy / Cyber / IPR Act amended from time to time.

MR.	Juber Abdul Momin	18UCS067
MR.	Sushant Suresh Patekari	18UCS071
MR.	Prajwal Nandkumar Patil	18UCS079
MR.	Shivshankar Yashwant Patil	18UCS081
MR.	Sumeru Shitalkumar Patil	18UCS083

ACKNOWLEDGEMENT

With great pleasure we wish to express our deep sense of gratitude to **Prof. K. S. Kadam** for his valuable guidance, support and encouragement in completion of this project report.

Also, we would like to take the opportunity to thank our head of department Dr. D. V. Kodavade for his co-operation in preparing this project report.

We feel gratified to record our cordial thanks to other staff members of the Computer Science and Engineering Department for their support, help and assistance which they extended as and when required.

Thank you,

MR.	Juber Abdul Momin	18UCS067
MR.	Sushant Suresh Patekari	18UCS071
MR.	Prajwal Nandkumar Patil	18UCS079
MR.	Shivshankar Yashwant Patil	18UCS081
MR.	Sumeru Shitalkumar Patil	18UCS083

ABSTRACT

During Covid19 pandemic, it became difficult for people to stay up-to-date with all related issues. Thus, it becomes obvious that if basic information on viruses can be provided to remote areas people can help them to take precautions and even prevent them from getting affected by providing answers to their questions quickly and accurately. Chatbots can be an effective way to
achieve this objective. This study presents the design of a sophisticated artificial intelligence (AI)chatbot based on semantic textual similarity providing information.

INDEX

Contents

1.Introduction

1.Introduction

A chatbot is a software application which can assume a role of a person virtually and carry out a conversation just like an ordinary person with the user. Chatbots have a wide range of applications in various fields. Chatbots can assist and guide an individual in many ways. Chatbots can be created as a static piece of code which is written to respond to specific inputs. A dynamic chatbot can be prepared by imparting some intelligence to it. This intelligence can be introduced using Artificial Intelligence (A.I) and Machine learning (M.L) algorithms. Such kind of chatbots can be more engaging and relevant to users compared to some chatbot with predefined set of rules. Now-a-days there are specific algorithms which can obtain semantic similarity between large texts , thus to find the answers to user queries by just find a matching question with the user query in the pre-existing database where the question is already mappedto an answer. Now coming to the point , keeping an eye on the current on- going pandemic situation, it would be a great idea to develop a health care chatbot able to answer questions related to coronavirus.This would help the concerned people to have urgent answers. The chatbot may keep them updated and even suggest preventive measures and precautions to take. A database can be obtained from any trusted source over the Internet such as an official healthcare related website for example -World Health Organization (W.H.O). Thus the database can be kept in sync with the source by updating it from time to time. Thus this chatbot can be really a great deal for users wanting a quickand accurate reply.

1.1 Problem definition

Since the appearance of the Coronavirus, it has spread across the globe becoming a severe pandemic. Thus during this period, great challenges have emerged forcing hospitals or healthcare staff to manage the flow of the high number of cases. Especially in remote areas where there is a paucity of medical professionals it is becoming more difficult . Thus, it becomes obvious that if basic information on viruses can be provided to remote areas people can help them to take precautions and even prevent them from getting affectedby providing answers to their questions quickly and accurately. This study presents the design of a sophisticated artificial intelligence (AI) chatbot based on semantic textual similarity providing information they need to know about the disease.

1.2 Aim and objective of the project

- To perform web scraping on the World health organization(W.H.O) web-site to create a database of frequently asked questions and their answers.

- Fine tuning Pretrained model on custom dataset for checking accuracyusing cosine similarity.

- Implementing basic functionality and creating User Interface where userscan enter queries and get relevant answers

1.3 Scope and limitation of the project

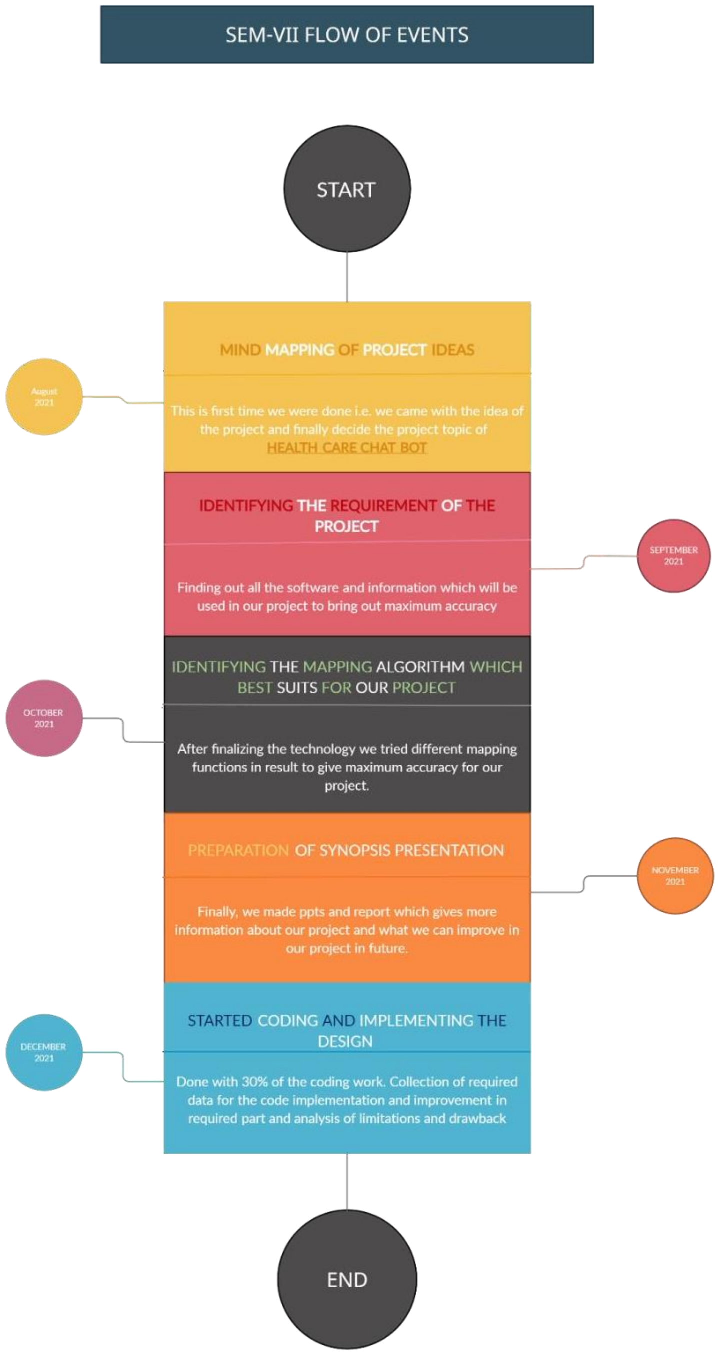
Scope:

- Health care chatbot receives covid -19 related questions from client and tries to understand the question and gives appropriate answer which will be helpful to the client to deal with their arised questions.Through the series of questions asked by the client it gives the appropriate solutions to diagnose the health condition of the patient.
- Health care chatbot is a powerful resource to resolve doubts about covid - 19 related questions.
- Health care chatbot gives direct answers to the queries instead of visit- ing several websites to get solutions for particular question.It will also recommend related questions which are frequently asked by the users.
- As we have given a platform to clients to get their appropriate solutions to their questions at one place so that chatbots become more user friendly and reliable as compared to others.
- It does this by converting a sentence or query asked by the client into ma- chine friendly query.Then going through the relevant data to find the nec- essary information and then answers in natural language .In other words it answers you as human does ,instead of giving different different solutions or list of different websites links that may have different different answers or suggestions.For example:if a user asks a query then it directly gives an- swer such as if user asks,What is corona? Then it simply gives information about corona and recommends some questions related to corona.

Limitations:

- The chatbot gives solutions only for the questions that are related to corona, not other diseases or other deficiencies.
- The chatbot gives answers to meaningful questions only.Chatbot cannot understand human general context.
- Our chatbot supports only the English language because the modules used by our chatbot are only trained for the English language.
- The chatbot answers the queries in a way that they have been taught another major limitation of the chatbot is decision making.

1.4 Timeline of the project



SEM-VII FLOW OF EVENTS

START

August 2021

MIND MAPPING OF PROJECT IDEAS

This is first time we were done i.e. we came with the idea of the project and finally decide the project topic of HEALTH CARE CHAT BOT

SEPTEMBER 2021

IDENTIFYING THE REQUIREMENT OF THE PROJECT

Finding out all the software and information which will be used in our project to bring out maximum accuracy

OCTOBER 2021

IDENTIFYING THE MAPPING ALGORITHM WHICH BEST SUITS FOR OUR PROJECT

After finalizing the technology we tried different mapping functions in result to give maximum accuracy for our project.

NOVEMBER 2021

PREPARATION OF SYNOPSIS PRESENTATION

Finally, we made ppts and report which gives more information about our project and what we can improve in our project in future.

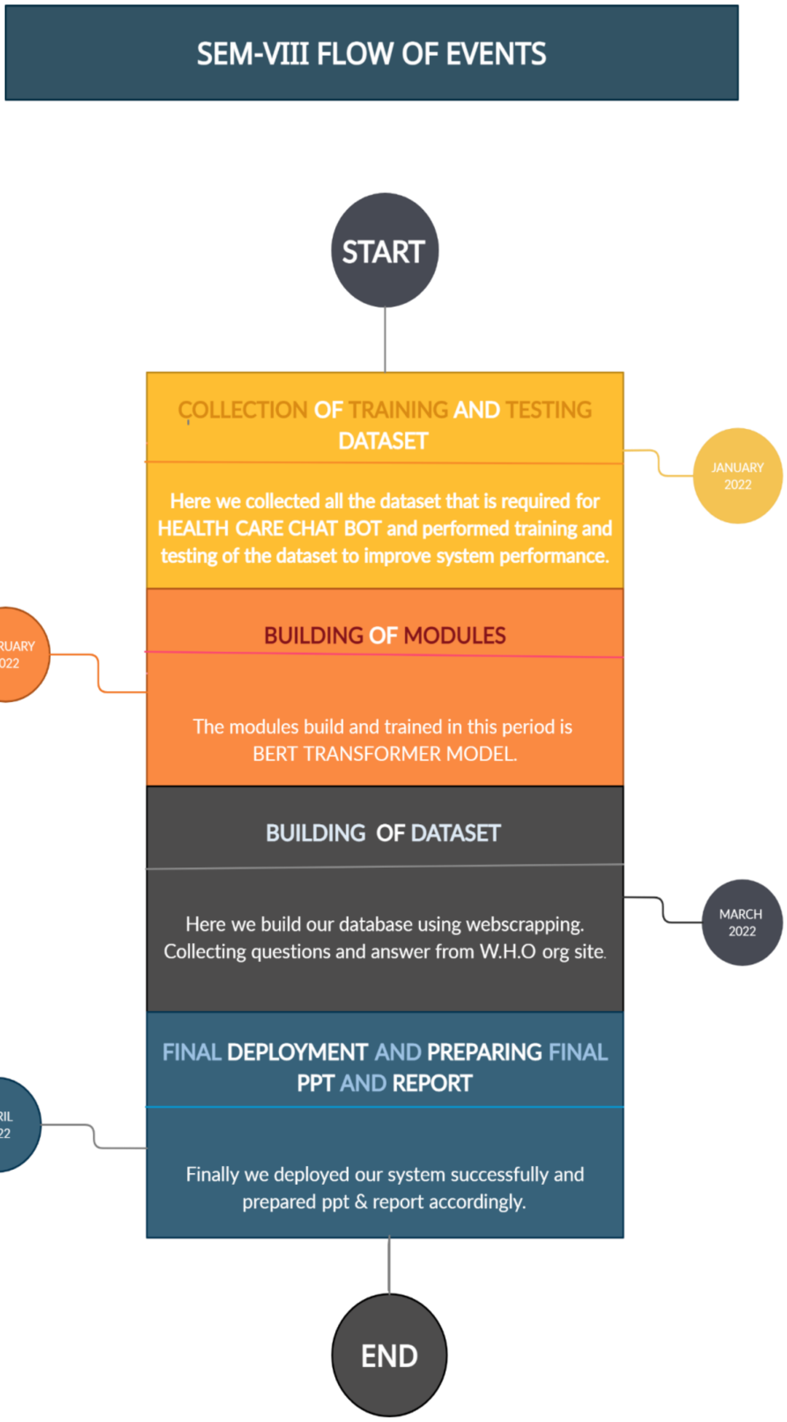
DECEMBER 2021

STARTED CODING AND IMPLEMENTING THE DESIGN

Done with 30% of the coding work. Collection of required data for the code implementation and improvement in required part and analysis of limitations and drawback

END

Figure 1:



SEM-VIII FLOW OF EVENTS

COLLECTION OF TRAINING AND TESTING DATASET

Here we collected all the dataset that is required for HEALTH CARE CHAT BOT and performed training and testing of the dataset to improve system performance.

BUILDING OF MODULES

The modules build and trained in this period is BERT TRANSFORMER MODEL.

BUILDING OF DATASET

Here we build our database using webscrapping. Collecting questions and answer from W.H.O org site.

FINAL DEPLOYMENT AND PREPARING FINAL PPT AND REPORT

Finally we deployed our system successfully and prepared ppt & report accordingly.

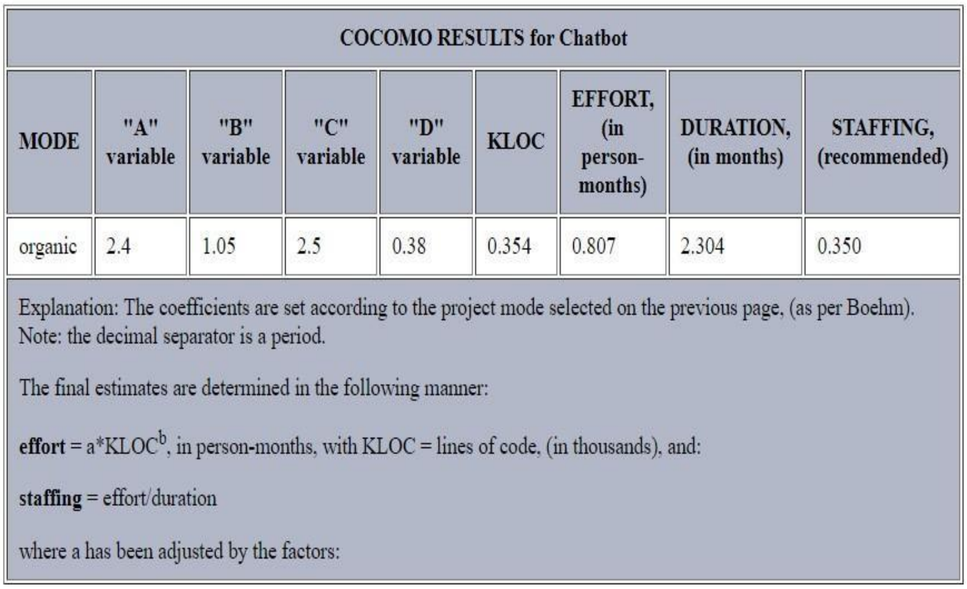
Figure 2:

1.5 Project Management Plan

Task	Duration	Start time	End time	priority
Topic selection	7 days	05-07-2021	12-07-2021	High
Topic discussion and Planning	7 days	13-07-2021	19-07-2021	High
Gathering information about project	7 days	20-07-2021	27-07-2021	High
Selecting and analyzing the problem statement	14 days	28-07-2021	14-08-2021	Medium
Finalization of project topic	7 days	15-08-2021	22-08-2021	High
Study on research paper	14 days	23-08-2021	6-09-2021	Medium
Documentation and synopsis	14 days	7-09-2021	21-09-2021	High
Requirement Analysis	7 days	22-09-2021	28-09-2021	High
System requirement	7 days	29-09-2021	4-10-2021	High
Module Identification and study	7 days	5-10-2021	12-10-2021	Medium
SRS Documentation and Presentation	14 days	13-11-2021	27-11-2021	Medium
Coding 30 percent and Data Collection	14 days	28-11-2021	11-01-2022	High
Selection of module	7 days	12-01-2022	18-01-2022	High
Implementation and model training	7 days	19-01-2022	26-01-2022	High
Database building and coding 70 percent	7 days	27-01-2022	03-02-2022	High
Implementation	7 days	4-02-2022	11-02-2022	High
Making GUI for project	7 days	12-02-2022	18-02-2022	High
Testing of module and Improvement	14 days	19-02-2022	03-03-2022	High
Coding 100 percent	7 days	04-03-2022	10-03-2022	High
Testing and Implementation 100 percent	7 days	11-03-2022	17-03-2022	High

Table 1:

1.6 Project Cost



COCOMO RESULTS for Chatbot								
MODE	"A" variable	"B" variable	"C" variable	"D" variable	KLOC	EFFORT, (in person-months)	DURATION, (in months)	STAFFING, (recommended)
organic	2.4	1.05	2.5	0.38	0.354	0.807	2.304	0.350

Explanation: The coefficients are set according to the project mode selected on the previous page, (as per Boehm). Note: the decimal separator is a period.

The final estimates are determined in the following manner:

effort = a*KLOC^{b}, in person-months, with KLOC = lines of code, (in thousands), and:

staffing = effort/duration

where a has been adjusted by the factors:

Figure 3:

2. Background study and literature overview

2. Background study and literature overview

2.1 Literature overview

- NLU Engine:
  Human language Understanding is a subpart of NLP (Human language Handle) which allows the system to accept the instinctive language or the conversational accent spoken for one user. The talkative language used by persons for day to day conversations is not as perfect as the precise
  language. It does not focus much on the terminology and the alphabet. Therefore, it enhances difficult for a system to learn the intent of the sen- tence. The recommendation taken from the consumer is in an unorganized text plan that cannot be assumed apiece order directly. It understands recommendations only in organized plans. The unorganized content re- ceived from the consumer is convinced to organize the plan by deriving important words and patterns from the consumer text utilizing the NLU methods. Persons are capable of understanding some mispronunciations, words pronounced the same as other, exchanged conversation, diminished form of words (like its for it is), argot dispute or phrases and again conver- sation that are not used casual glossary but endure in normal dialogues. NLU techniques authorizes bureaucracy to recognize these twerks if the consumer forms the use of ruling class while conversing accompanying the chatbot, so concerning form the consumer feels that the discourse is taking place in the middle from two points between two persons and not between a human and a bot. NLU systems do not straightforwardly understandthe significance of the consumer sentences. It includes a series of processes to determine the real intent of the sentence. To appreciate a complete sen- tence, the NLU scheme needs to accept each discussion of that sentence. The initial task is the separation of the sentences into individual conver- sation. Next, to believe the word, the system needs to understand the syntax of the sentence. This may be finished by being aware of the parts of speech of each word in that sentence. Here comes the POS (Parts-Of-

Speech) tagger into picture. After being aware of the semantic weightage of each word, all of them are parsed to know the dependency with them. This is the most important step in what way legal order accompanying the best dependency is extracted, from which the determination of the system may be known. It is not possible that the information base would hold the exact sentence that the consumer has shipped. Its power contains a sentence with the alike determined but accompanying various words used in it. To equal these types of equivalent sentences, synonym determina- tion and sentence matching are necessary. The various tasks expected to be implemented under the NLU Generator and the means to do the un- changing have happened explained further.

1964 by Daniel Bobrow for his PhD dissertation at MIT, is individual of the earliest popular attempts at natural-language understanding by a computer. Eight years later John McCarthy coined the term machine intelligence, Bobrow's dissertation (named Human language Input for a Computer Problem Answering System) showed by means of what a computer could believe simple natural language recommendation to solve al- gebra word questions.

In 1982 Gary Hendrix made Symantec Corporation initially as a com- pany for developing a human language interface for database queries on personal computers. Still, with the coming of mouse-driven graphical consumer interfaces, Symantec changed management. A number of added commercial efforts were started around the synchronal

-
- Word Segmentation:
  Separation, further referred to as tokenization, is the process of dividing text into tinier and significant holes. These parts may be paragraphs,sentences, sections, phrases, words or letters. The minimal part is theanswers. Discussion segmentation is the dividing of sentences into individual conversation separated by blank rooms. The tokenized parts of the sentences are named as tokens. The tokenizers split the sentences into dispute and punctuation marks as liberated parts. The most commonly used tokenizer is of scope type, that is it splits the sentences into wordsat the blank scopes. It is further necessary that the tokenizer should deal with contractions, acronyms, dates, numbers in having ten of something layouts, etc., that cannot split at punctuations and blank scopes, as they will escape their meaning if finished so.

  Mohammed Javed et al. [1] [2015] elucidated an arrangement to imple- ment dispute separation. He projected in welcome treasure to reckon completeness spaces in the sentences. The type rooms hold all types of break between individualities.They hold the gap between letters, punc- tuations and allowable order. The algorithm functions on the supportof the amount of breach or purity outlook middle from two points each part in the sentence. Later the forecast of individuality capacity, an aver- age of the break is calculated to see the mean average between figures in the sentence. This average breach distance is before used to the sentence that is to say anticipated separately. The places where the personality scope is more than the average type outlook are pronounced to anticipate the points of tokenization. The break middle from two points dialogue is steadily apart from the average breach and then tokenization takes place at the blank spaces between conversation in the sentences.

Naeun Lee et al. [2] [2017]projected the exercise of discussion segmen- tation using NLTK. Everyday Style ToolKit (NLTK) is a python tool that caters to support duties for NLP. It has inbuilt tokenizers. Consumers need to significance the whole and use the necessary type of tokenizer thatis present in the form of functions. The NLTK involves a wide range of tokenizers which are as follows: standard, message, discussion, clas- sic, lowercase, N-gram, pattern, magic words for entry, course, etc. The most usually used tokenizer is the discussion-punkt tokenizer that splits the sentences at the blank rooms. The veracity, speed and efficiency of the NLTK tokenizers is commendable. Again, it does not demand some algorithm exercise as the package executes them at the backend.

- POS Tagging:

POS Tagging is the process of designating semantic annotations to individ- ual words in the sentences. These annotations involve the Parts-Of-Speech
Tags. They mean the grammatical significance of the word in the sentence based on the dependency of that discussion with additional words in that phrase, clause, sentence, article, etc. The ordinary POS tags are proper nouns, action words, pronouns, etc. There are a number of ways that may be used to act as POS Tagging.

Jerome R. Bellegarda [3] [2010] projected a system named latent anal- ogy for POS Tagging. In this algorithm, latent semantic mapping (LSM) technique is used. It requires the preparation of the available corpus. The LSM claims a feature space of the prepared entirety that has existed tagged. Immediately, new sentences are given to the LSM for tagging and the study is acted so as to determine the sentences from the training data that are tightest to the test sentence. This is called the sentence neigh- borhood. Sentence neighborhood holds real for two sentences if they share the unchanging intent matter. Earlier the resolute matching sentences are found from the prepared data, the POS tags attributed to those sentences are then mapped to the test sentences.

- Dependency Parsing:
  A dependency parser is secondhand to authorize the relationship between words in a sentence with the semantic tags joined to it. It is the next
  step after parsing. A dependency tree or diagram is conceived for each sentence. This sapling is called the parsing tree or the dependency tree. There are many ways by which the parsing may be executed.

  Bo Chen [4] [2011]projected a form for implementing the dependency tree. It originally discovers the dependencies between the words in the sentence.

  Each word is checked for allure connection or reliance with the other dis- cussion. Legal order accompanying the highest dependency is picked and expected at the root. The other dispute with a connection with the root node is ascribed to it as the infant knots. This keeps on continuing asfar as all the disputes are established in the tree. The tree form of the sentence is named the dependency parser tree. The dependencies with the words are discovered by utilizing the POS tags.

- NLG Engine:
  NLG acts the reverse task of NLU. It is the process of changing the system that causes results into human language representations that may be clear by
  the consumer. In addition, NLG is the process of creating text/speech from whole generated patterns. The results created by the system are in the organized layout because they may be clear and processed by the system. NLG shows the system information base in an everyday or conver- sational expression likeness that may be obvious for one user. There may be a number of habits at which point in which an unchanging sentencecan be said. The sentences can have two voices that are alive or lifeless. Further, skilled maybe likeness 'tween two sentences, but their power in- cludes the custom of synonyms. Therefore, while providing an answer to the consumer, the NLG unit needs to calculate all the potential to define the unchanging sentence, and therefore select the most appropriate indi- vidual. NLG appliance also acts as an order of tasks to create sentences. The primary task is to decide the content. It includes the collection of reactions expected given to the consumer. This step ends the appropriate content (or set of dispute) that concedes the possibility of being present in the sentence. Too, it deals with the position of mandate in the sentences established allure Mail service Tag (placements of deponents, nouns, ad- juncts, prepositions, etc.). As a whole, this step handles the institution of an elementary sentence right from the choice of dispute to their place- ment in the sentence. The next task is the choice of sentences. As earlier said, skilled may be a difference of sentences that may be used to express the alike position, this step handles the collection of the appropriate sen- tence, that is highest in rank for that instance. The sentences captured into concern for potential are in their abstract layout and are not perfect sentences. They require the adding of alphabet rules to form bureaucracy grammatically correct. This portion checks the semantic accuracy of sen- tences

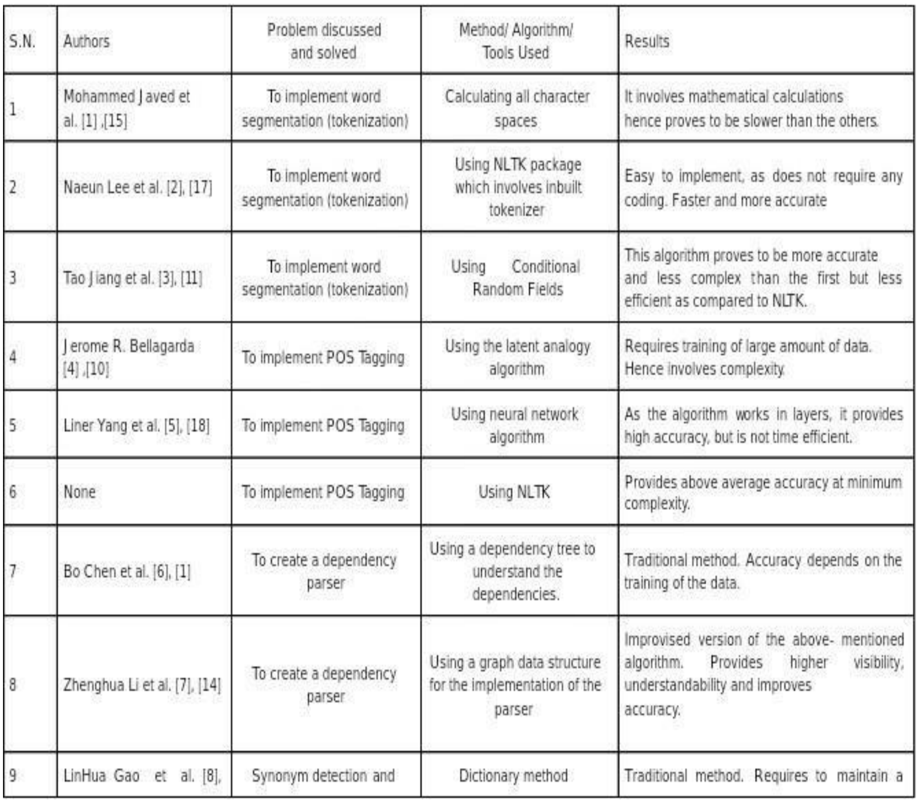
  established by the alphabet rules outlined by the system. Last and ultimate main is the makeup check, in what way the sentence produced from the previous steps is checked upon for allure correctness. This step validates the accuracy of the sentence.

Sachin S. Gavankar et al. [5] [2017] projected the enthusiastic decision tree algorithm for prediction. This type of decision tree is the made-up version of the usual decision tree. It constitutes this sapling at runtime, located on the consumers queries and keeps updating the shrub on new consumer ideas. Deal with its active for affliction guess. In this algorithm, the symptoms discovered in the consumer query are additional as child growth to the root node. The nodes keep on getting additional for new symptoms discovered. Further for every symptom, the algorithm checks for the second syndrome that has the highest incident with the former symptom and asks the user for that syndrome. If he announces agreed, therefore the system traces that path to check for the disease present at the root node. This will maintain iterating for all users and the tree keeps getting renewed for new entries or traces the path usable.

- Synonym and Pattern Recognition: For information retrieval, despite how big our data is, no sentence shipped by the consumer can be perfectly the same to some sentence in the table. But there can be sentences with the same intent. Later understanding the intent of the consumer sentence, thetable is inspected for a sentence with the same determination. The doubled sentences have a difference of words which are used to express theunchanging content. They use alternative conversation or synonyms. Thismakes analogue discovery unavoidable for the system. Synonyms for aparticular word concede the possibility of being domain independent or domain contingent. Domain independent synonyms are synonyms for adiscussion in the entire vocabulary. But domain-dependent synonyms aresynonyms for a discussion in that particular domain only. There are vari-ous algorithms used for the discovery and extraction of synonyms.

Sijun Qin [6] [2015] projected a feature selection pattern for synonym extraction. In this place method, between all the parts of speech tags, words having the tags as nouns, infinitives and adjuncts are marked as positive tags and the possible choice as negative tags. The opposition for each feature (discussion) is before carried out by utilizing the POS tags. If the overall feature opposition is definite, then it may be identified absolutely. All the positive features are then arranged together and the synonyms discovery for the group of looks will be relatively strong, as the whole passage is inspected for its synonymic meaning. The analogue sets that are derived for that section of features are before determined for news gain. The individual with the capital news gain is the most powerful analogue extracted.

2.2 Critical appraisal of other people’s work



S.N.	Authors	Problem discussed and solved	Method/ Algorithm/ Tools Used	Results
1	Mohammed J aved et al. [1] ,[15]	To implement word segmentation (tokenization)	Calculating all character spaces	It involves mathematical calculations hence proves to be slower than the others.
2	Naeun Lee et al. [2], [17]	To implement word segmentation (tokenization)	Using NLTK package which involves inbuilt tokenizer	Easy to implement, as does not require any coding. Faster and more accurate
3	Tao J iang et al. [3], [11]	To implement word segmentation (tokenization)	Using Conditional Random Fields	This algorithm proves to be more accurate and less complex than the first but less efficient as compared to NLTK.
4	J erome R. Bellagarda [4] ,[10]	To implement POS Tagging	Using the latent analogy algorithm	Requires training of large amount of data. Hence involves complexity.
5	Liner Yang et al. [5], [18]	To implement POS Tagging	Using neural network algorithm	As the algorithm works in layers, it provides high accuracy, but is not time efficient.
6	None	To implement POS Tagging	Using NLTK	Provides above average accuracy at minimum complexity.
7	Bo Chen et al. [6], [1]	To create a dependency parser	Using a dependency tree to understand the dependencies.	Traditional method. Accuracy depends on the training of the data.
8	Zhenghua Li et al. [7], [14]	To create a dependency parser	Using a graph data structure for the implementation of the parser	Improvised version of the above- mentioned algorithm. Provides higher visibility, understandability and improves accuracy.
9	LinHua Gao et al. [8],	Synonym detection and	Dictionary method	Traditional method. Requires to maintain a

Figure 4:

2.3 Investigation of current project and related work

Transformers are state-of-the-art machine learning algorithms which are compatible with Pytorch, Tensorflow and JAX. Transformers provide a large number of pretrained models which helps in reducing compute costs , carbon footprints and saving time from training a model from scratch.Transformers can be fine-tuned on custom datasets based on the requirements. Some ba-sic applications of transformer models are - text classification, information ex- traction, question answering, summarization, translation , image classification, speech recognition , OCR etc. BERT (Bidirectional Encoder Representation from transformers) is a recent paper published by researchers at Google AI language.It has caused a stir in the Machine learning community by presenting state-of-the-art results in a wide variety of NLP tasks, including Question Answering, Summarization, Text generation , text classification, translation etc.BERT makes use of Transformer, an attention mechanism that learns con- textual relations between words in a text.BERT provides sentenceTransformers which is a Python framework for state-of-the-art sentence text and image embedding.This framework can be used to compute sentence/text embedding for more than 100 languages.These embedding can be compared e.g. with cosine- similarity to find sentences with a similar meaning.This can be useful for seman- tic textual similarity,semantic search or paraphrase mining.Finally , it is easy to fine-tune your own models.

3.Requirement Analysis

3.Requirement Analysis

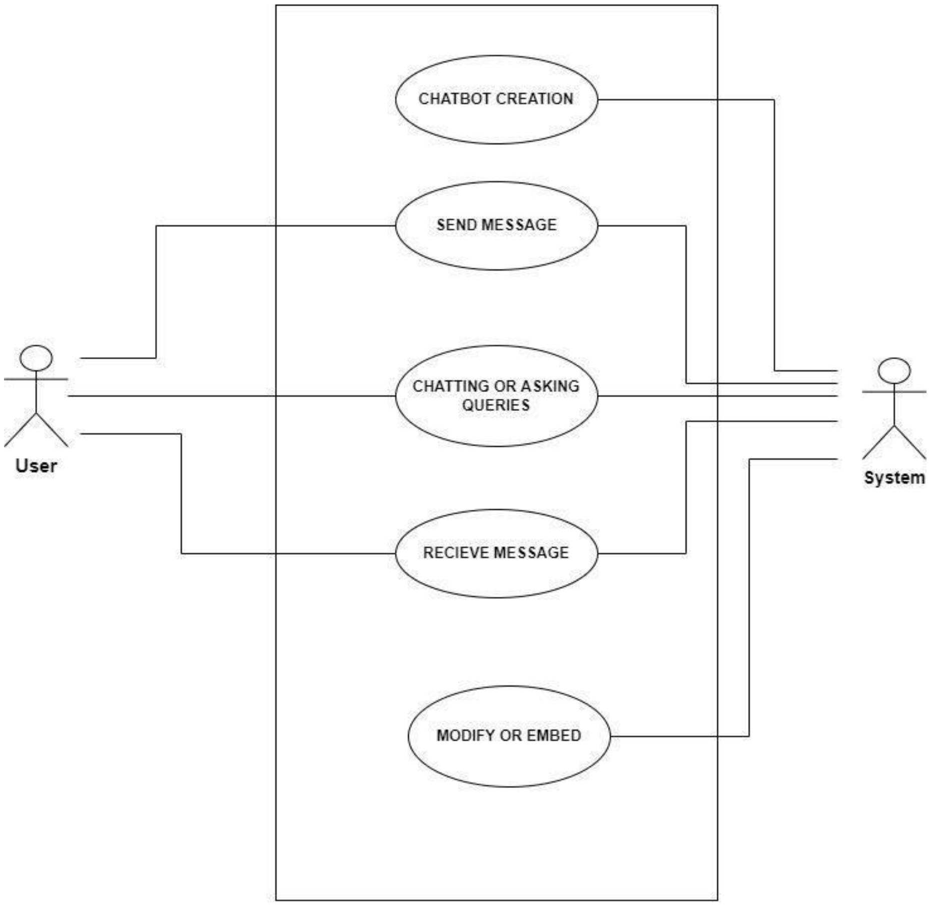
3.1 Requirement Gathering

- As a user, I want to access chatbot user interface
- As a user , I want to start asking problems/questions related to Health problems through the user interface provided by user interface in chat bot.
- As a user, I want the next related questions asked by me and deep analysis related my question
- USER STORIES
- Health Care Chatbot helps improve user experiences by providing answers during crucial decisions.
- In addition to understanding and interacting within conversations, health care chatbot software has Cosine similarity to analyze the context of a conversation.
- It can identify the intent of a question to provide an accurate answer and suggest options to confirm or resolve the issue.
- The Health care chatbot can capture, read and process large amounts of data to gain insights from relevant data and to quickly solve user problems related to Health
- Healthcare chatbot software should continuously add new frequently asked questions by analyzing conversations
- It is also useful if data and context can be stored over several channels.

3.2 Requirement Specification

Specification	Action Performed	Input/output	Requirement
Purpose	This System aimsto study and ex-plore healthcare related Solutions by utilizing methods and techniques sentece transformer model.With the continous rise in healthy related problems get automatically updated.	NONE	NONE
Input/Output	NONE	Input:-Ask Questions through user interface provided.Click related questions shown on user interface. Output:-Answer is given by the chatbot to question asked by user.	NONE
Functional Requirement	Data Collection. Identify the intent of question.Suggest solution to problem.Suggest options to Given question.	NONE	NONE
Facilities Required	For this project to work we require a python environment.The System must have minimum 2GB of RAM.Also we require a stable internet connection.	NONE	NONE

3.3 Use case Diagram



USE CASE DIAGRAM FOR QUESTION ANSWER BASED HEALTH CARE CHATBOT

Figure 5:

4. System Design

4. System Design

4.1 Architectural Design

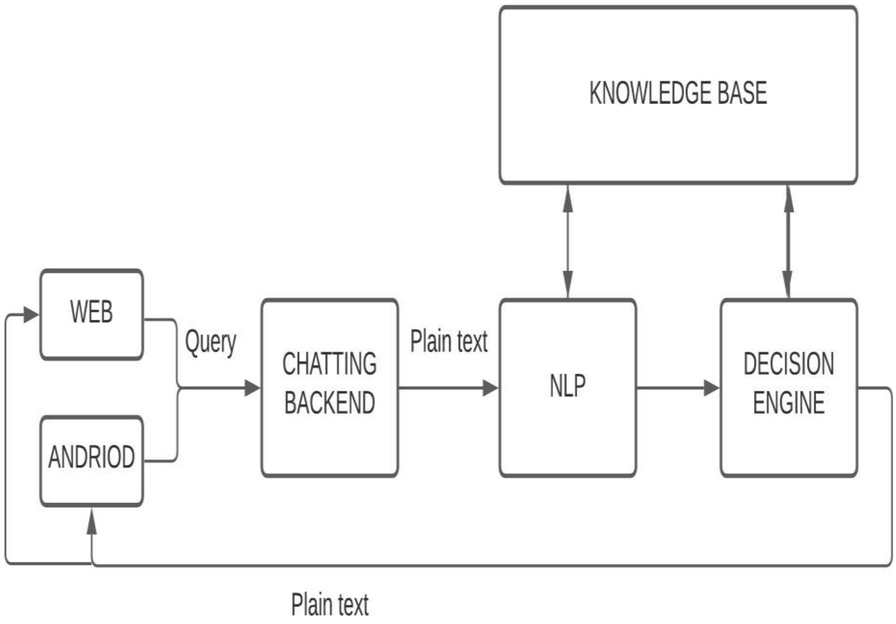


Figure 6:

4.2 User Interface Design

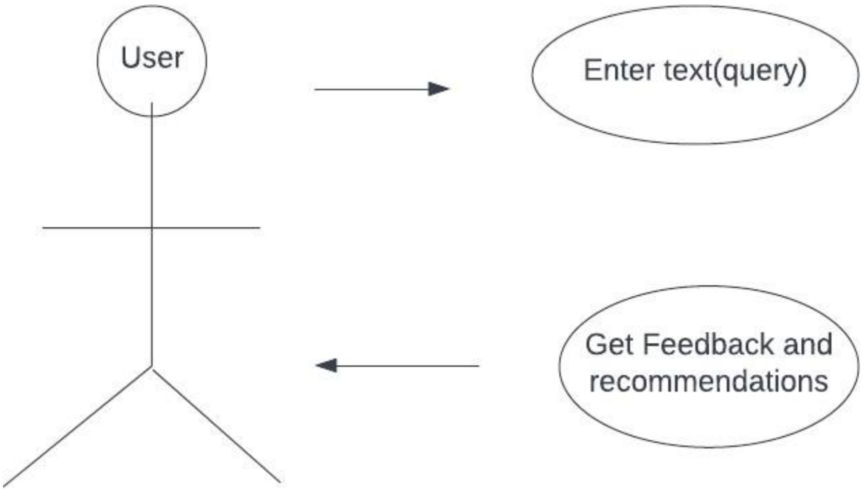


Figure 7:

4.3 Algorithmic description of each module

- Web scraping module -
  1. START
  2. Import essential modules
  3. Assign website link
  4. Create array of webpages to be traversed
  5. Create BeautifulSoup object
  6. Loop over webpage array and extract Question-Answers usingHTMLtags
  7. Save to CSV file
  8. STOP
- Model training module -
  1. START
  2. Import modules
  3. Declare and Initialize dataset and model
  4. Set hyperparameters
  5. Split the dataset in training, testing and validating
  6. Execute model.fit()
  7. Save the model
  8. Execute evaluator to check accuracy
  9. END
- Main application module -
  1. START
  2. Import modules
  3. Declare and initialize model,tokenizer
  4. Read database
  5. Obtain the embeddings of database
  6. Get user query
  7. Embed user query
  8. Obtain score to check similarity
  9. Assign answer of question with maximum score
  10. Assign related questions with 2nd and 3rd highest scores
  11. Display the response
  12. STOP

4.3.1 Dataflow Diagram

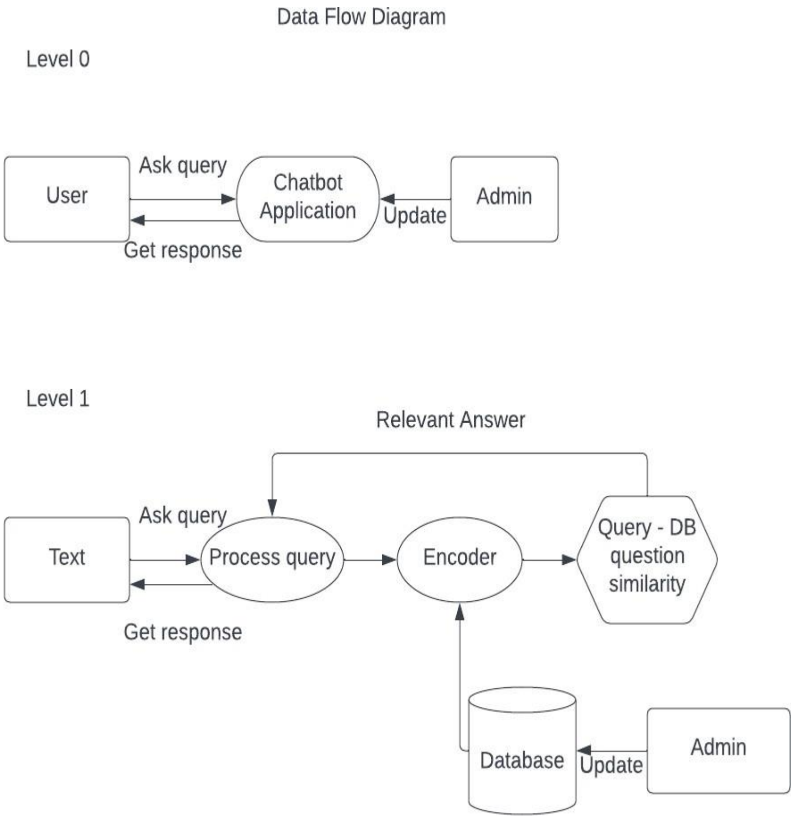


Figure 8:

4.3.2 Sequence Diagram

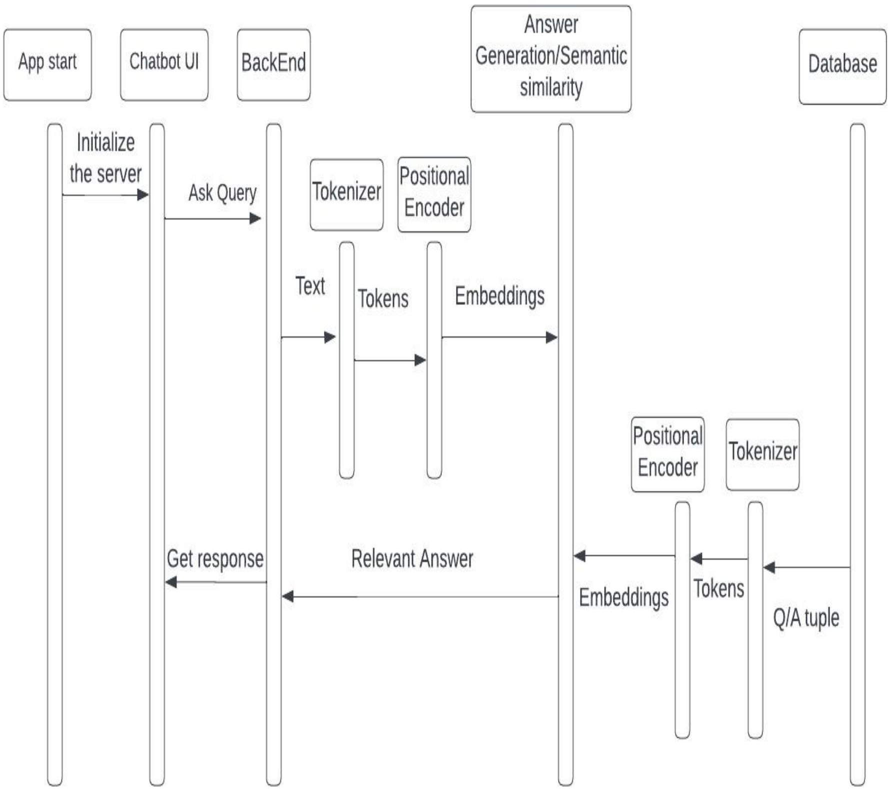


Figure 9:

4.3.3 Activity Diagram

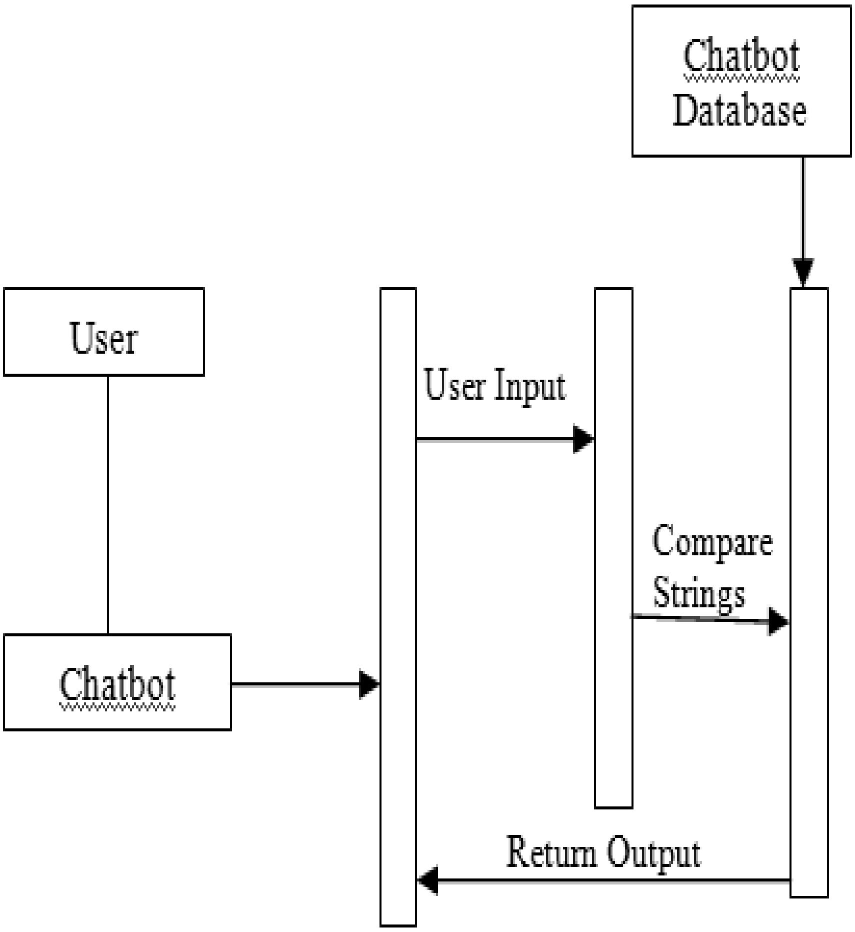


Figure 10:

4.3.4 Deployment Diagram

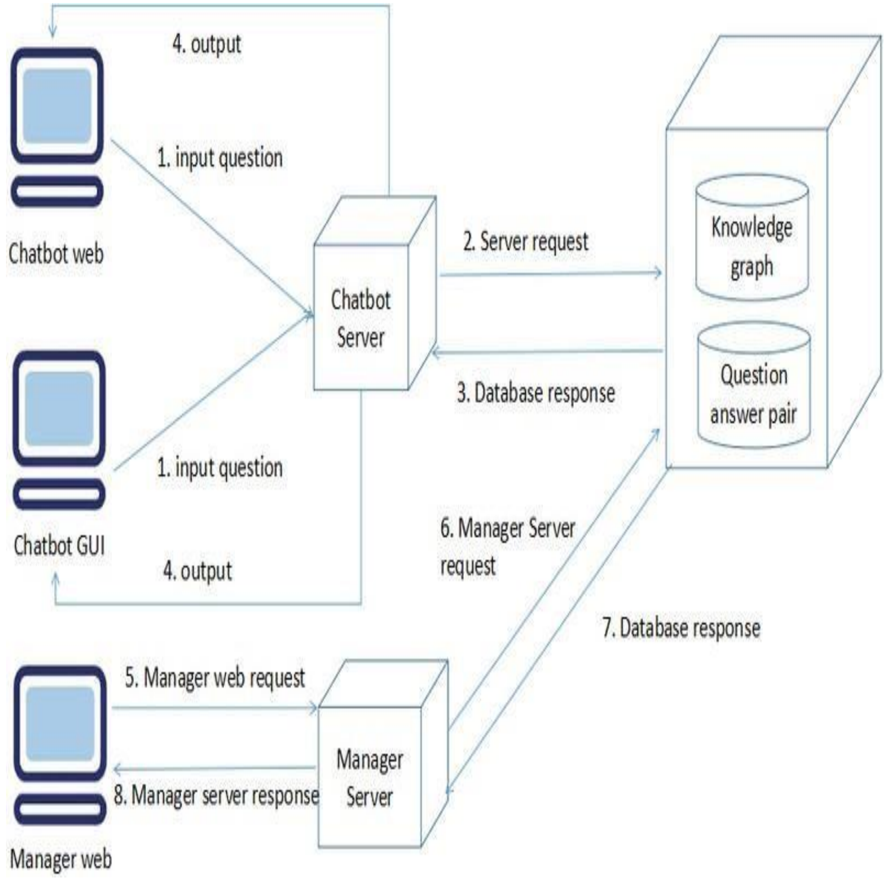


Figure 11:

5.Implementation

5. Implementation

5.1 Environmental Setting for Running the Project

The project is by and large based on Python3 and executed in the Linux environment. For running the project, a python virtual environment is to be created. Following is the command to create a virtual environment at user spec- ified project path.
Command -

```
python3 -m venv env
```

Once the virtual environment is set up , activate it by usingCommand -

```
source env/bin/activate
```

Now all the required libraries and dependencies can be installed using pip com-mandCommand -

```
pip install package-name
```

Required libraries and packages -

- Web scraping module requirements
  - requests
  - csv
  - re
  - pandas
  - bs4
- Model training module requirements
  - torch
  - math
  - sentenceTransformers
  - logging
  - datetime
  - sys
  - os
  - gzip
- Main Application module requirements
  - transformers
  - pandas
  - torch
  - flask

5.2 Detailed Description of Methods

System consists of three main modules -

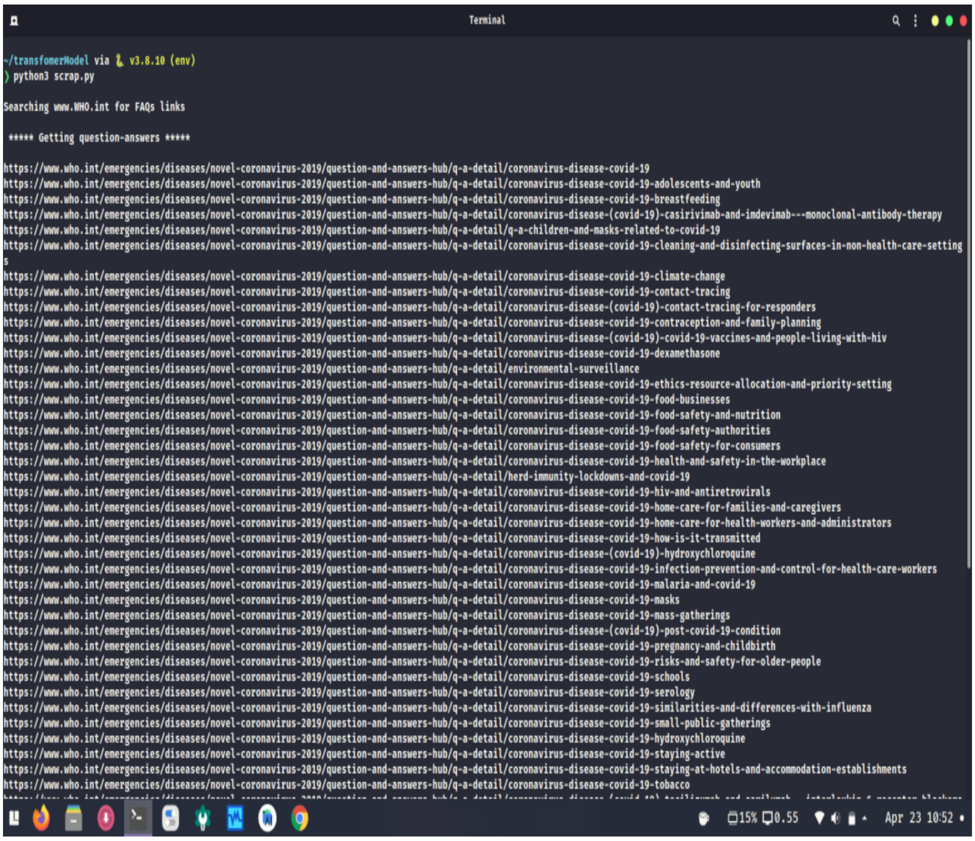
1. Web scraping
2. Model Training
3. Main Application

Each modules contains specific number of functions/methods to meet all func-tionalities specified in requirement specifications

1. Web scraping module - This module contains only one function. This function performs web scraping on W.H.O's official website. The function extracts frequently asked questions and their respective answers , formats the html and saves it in a .csv file.
2. Model training module - As the name suggests , the model is trained in this part or more specifically pre-trained model - 'distilbert-base-uncased' is fine-tuned on a custom dataset of choice (for example - stsbenchmark.tsv dataset.)
3. Main application module -
   - mean-pooling method - This method takes two parameters: model- output and attention-mask. The method takes the average of all tokens (word embeddings) stored in model-output. This gives us a fixed 768 dimensional output vector independent of how long our input text was.
   - encode-method - This method takes the text to encode. A tokenizer from the pre-trained transformer model is used to tokenize the query. Mean pooling is applied to tokens and then normalized.
   - Get-bot-response-method - This method involves the implementation of the flask framework to create a simple web Graphical user interface. It receives the user query , encodes the query and calculates the scores based on cosine similarity. The mapping getting the maximum scoreis selected as the answer.Similarly related questions are presented to the user based on the scores.

5.3 Implementation Details

- Implementation of web scraping module - The required libraries are im- ported at the start. A direct link to WHO frequently asked questionsis stored in a variable. bs4 (BeautifulSoup) is a library in python which makes it easy to scrape information from websites and helps in extracting the data from HTML and XML files.This library needs to be downloaded externally as it does not come readily with the Python package. To get HTML documents from the URL , use request.get() method bypassing the URL to it. Then a parse tree object is created by passing theHTML document extracted and python built-in HTML parser. Using specific tags to extract the links from the BeautifulSoup object. Get the actual URLs from the form all anchor tag objects with get() method andpassing href argument to it.Using find-all method get questions and answers and store them in a dataframe. Finally the dataframe is convertedto csv file. Thus the database for the question-answer model is ready touse.

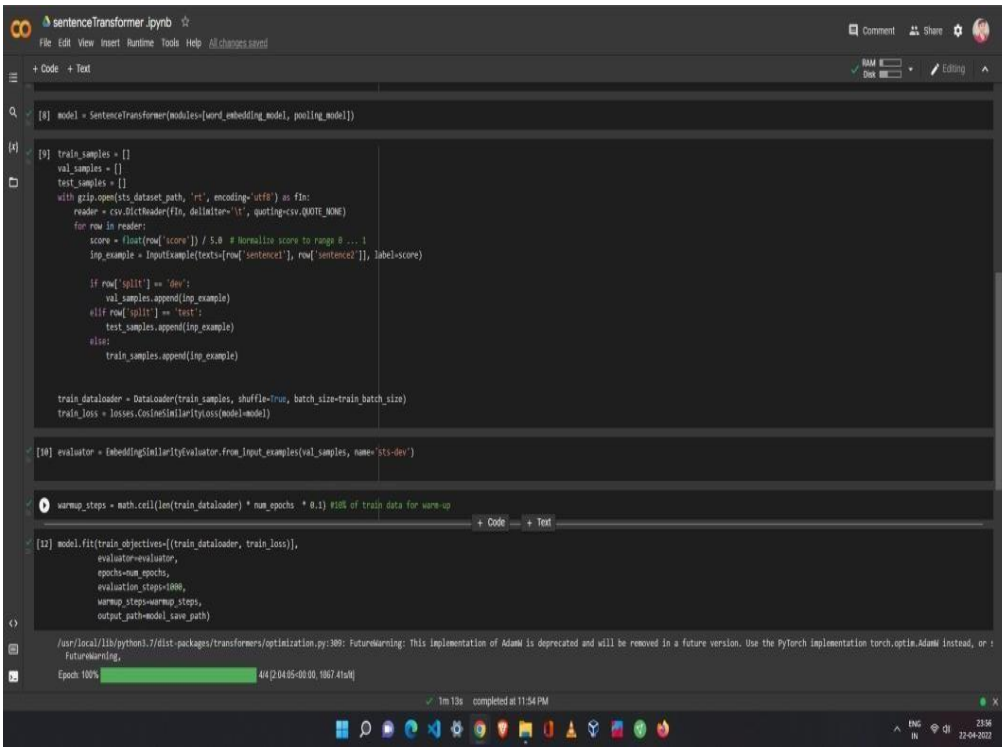


```
~/transfomerModel via 🐍 v3.8.10 (env)
❯ python3 scrap.py

Searching www.WHO.int for FAQs links

 ***** Getting question-answers *****

https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-adolescents-and-youth
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-breastfeeding
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-(covid-19)-casirivimab-and-imdevimab---monoclonal-antibody-therapy
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/q-a-children-and-masks-related-to-covid-19
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-cleaning-and-disinfecting-surfaces-in-non-health-care-setting
s
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-climate-change
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-contact-tracing
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-(covid-19)-contact-tracing-for-responders
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-contraception-and-family-planning
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-(covid-19)-covid-19-vaccines-and-people-living-with-hiv
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-dexamethasone
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/environmental-surveillance
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-ethics-resource-allocation-and-priority-setting
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-food-businesses
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-food-safety-and-nutrition
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-food-safety-authorities
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-food-safety-for-consumers
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-health-and-safety-in-the-workplace
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/herd-immunity-lockdowns-and-covid-19
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-hiv-and-antiretrovirals
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-home-care-for-families-and-caregivers
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-home-care-for-health-workers-and-administrators
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-how-is-it-transmitted
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-(covid-19)-hydroxychloroquine
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-infection-prevention-and-control-for-health-care-workers
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-malaria-and-covid-19
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-masks
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-mass-gatherings
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-(covid-19)-post-covid-19-condition
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-pregnancy-and-childbirth
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-risks-and-safety-for-older-people
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-schools
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-serology
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-similarities-and-differences-with-influenza
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-small-public-gatherings
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-hydroxychloroquine
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-staying-active
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-staying-at-hotels-and-accommodation-establishments
https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-tobacco
```

Figure 12:

Implementation of training model - Install the sentence-transformer python package and import all necessary modules.SenteneTransformers was de- signed in such a way that fine-tuning your own sentence / text embedding models is easy. For sentence / text embeddings ,we want to map a variable length input text to a fixed sized dense vector. Then feed the input sentence into a transformer network like BERT. BERT produces contextu-alized word embeddings for all input tokens. Using mean-pooling gives a fixed 768 dimensional output vector independent of input size. Sts Bench- mark dataset is used to fine-tune the pre-trained model. The dataset is split into train-samples, dev-samples, test-samples.Then using dataloader we can shuffle and create batches with specific batch-size. To calculate train loss , cosineSimilarityLoss is used.Then set the parameters and fit the model.

•

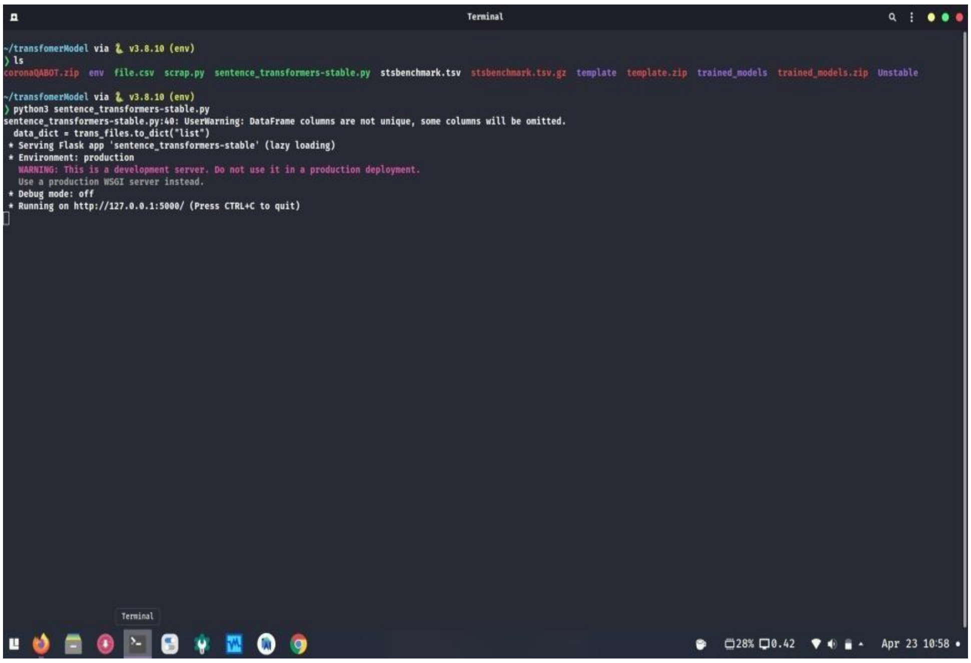


sentenceTransformer.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing

```
[8] model = SentenceTransformer(modules=[word_embedding_model, pooling_model])

[9] train_samples = []
    val_samples = []
    test_samples = []
    with gzip.open(sts_dataset_path, 'rt', encoding='utf8') as fIn:
        reader = csv.DictReader(fIn, delimiter='\t', quoting=csv.QUOTE_NONE)
        for row in reader:
            score = float(row['score']) / 5.0  # Normalize score to range 0 ... 1
            inp_example = InputExample(texts=[row['sentence1'], row['sentence2']], label=score)

            if row['split'] == 'dev':
                val_samples.append(inp_example)
            elif row['split'] == 'test':
                test_samples.append(inp_example)
            else:
                train_samples.append(inp_example)


    train_dataloader = DataLoader(train_samples, shuffle=True, batch_size=train_batch_size)
    train_loss = losses.CosineSimilarityLoss(model=model)

[10] evaluator = EmbeddingSimilarityEvaluator.from_input_examples(val_samples, name='sts-dev')

warmup_steps = math.ceil(len(train_dataloader) * num_epochs  * 0.1) #10% of train data for warm-up

[12] model.fit(train_objectives=[(train_dataloader, train_loss)],
               evaluator=evaluator,
               epochs=num_epochs,
               evaluation_steps=1000,
               warmup_steps=warmup_steps,
               output_path=model_save_path)

/usr/local/lib/python3.7/dist-packages/transformers/optimization.py:309: FutureWarning: This implementation of AdamW is deprecated and will be removed in a future version. Use the PyTorch implementation torch.optim.AdamW instead, or :
  FutureWarning,
Epoch: 100% 4/4 [2:04:05<00:00, 1067.41s/it]
```

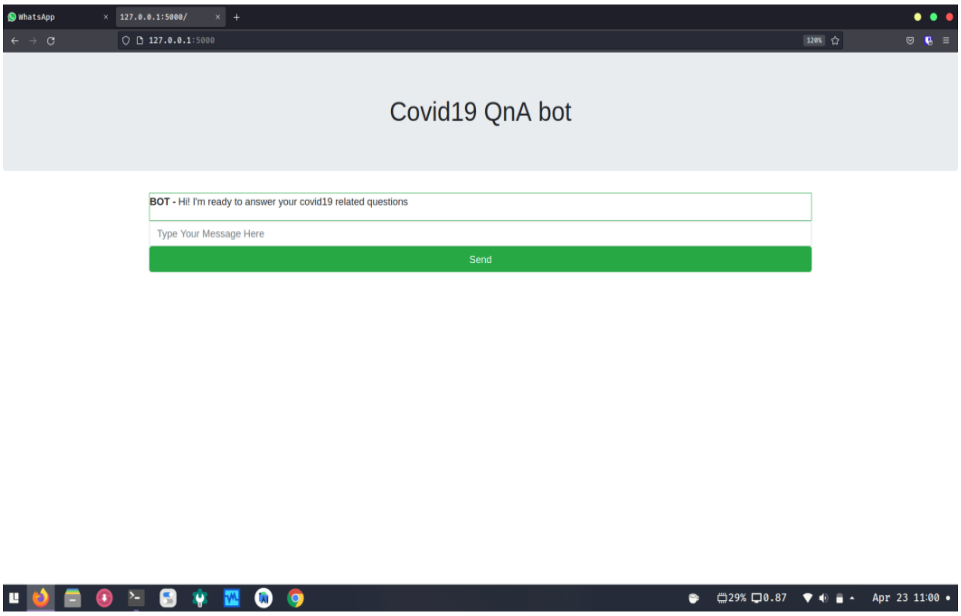
Figure 13:

- Implementation of Main application - All necessary modules are imported . A tokenizer is used to make tokens from input text. The tokenizer is selected using autoTokenizer from a pretrained model. Divide the func- tionality in different methods. Create a mean-pooling method for perform- ing pooling on the tokens created by tokenizer.Create a encode methodto obtain token embeddings. Obtain encodings of the question-answer database file. Define the user response function to obtain the query. Take the query from the user. Encode the query accordingly then obtain scores using cosine similarity to determine the similarity between questions in the database and user query. Select the answer mapped to the question with maximum score. Obtain related questions based on scores. Finally display the response to the user’s screen.



```
~/transfomerModel via 🐍 v3.8.10 (env)
❯ ls
coronaQABOT.zip  env  file.csv  scrap.py  sentence_transformers-stable.py  stsbenchmark.tsv  stsbenchmark.tsv.gz  template  template.zip  trained_models  trained_models.zip  Unstable

~/transfomerModel via 🐍 v3.8.10 (env)
❯ python3 sentence_transformers-stable.py
sentence_transformers-stable.py:40: UserWarning: DataFrame columns are not unique, some columns will be omitted.
  data_dict = trans_files.to_dict("list")
 * Serving Flask app 'sentence_transformers-stable' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Figure 14:



Covid19 QnA bot

BOT - Hi! I'm ready to answer your covid19 related questions

Figure 15:

6. Integration and Testing

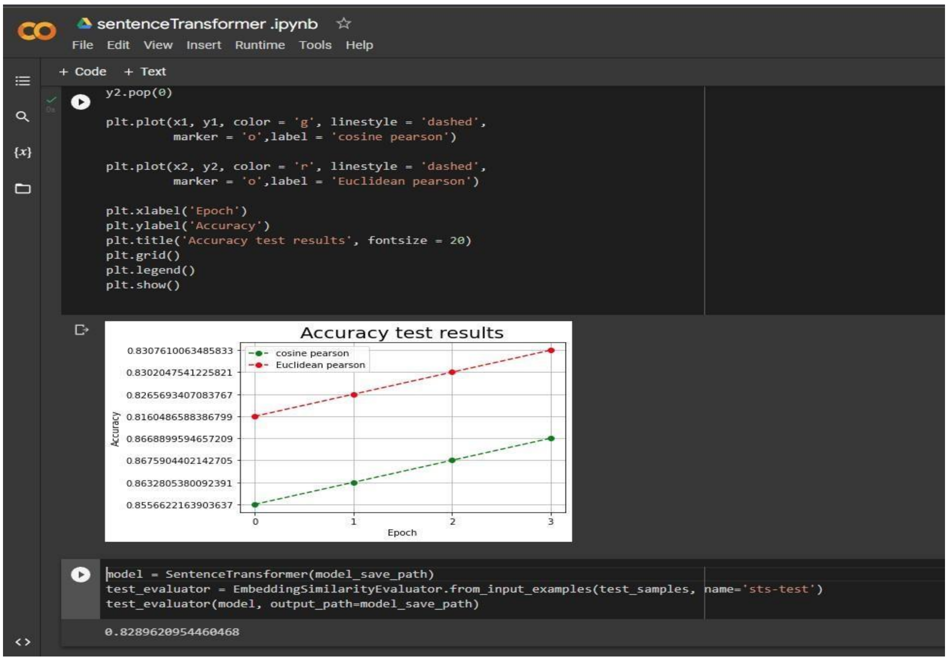
6. Integration and Testing

The chatbot application has different modules which need to be integratedto create a fully functioning software application.The web scraping module, transformer module,template module and flask module needs to be integrated . Web scraping module can be executed from the transformer module to update the database according to the websites frequently asked questions. Flask is a python framework to create GUI for applications. Flask can be easily integrated into the application using import statement.We define the app route to point to html and css file in template directory thus file to render the chatbot interface on a browser can easily be integrated. Flask deploys the server and handles the user requests coming from their web browsers. Thus once a request is received it can be processed with the application logic and then a response is sent back to the user. All these modules work in coordination with each other to provide the basic functionalities of the application.

7. Performance Analysis

7. Performance Analysis

To check the performance of the application , accuracy of the trained model is evaluated. Below is the graph plotted with accuracy against each epoch. Cosine pearson and Euclidean pearson are the two metrics used to measure the accuracy. Manually testing on the end product can be performed by checking the responses to the user query. Entering a query and submitting it to the application server and checking the relevance of the response answer can be a simple but efficient method to measure the accuracy.



```
y2.pop(0)

plt.plot(x1, y1, color = 'g', linestyle = 'dashed',
         marker = 'o',label = 'cosine pearson')

plt.plot(x2, y2, color = 'r', linestyle = 'dashed',
         marker = 'o',label = 'Euclidean pearson')

plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.title('Accuracy test results', fontsize = 20)
plt.grid()
plt.legend()
plt.show()
```

```
model = SentenceTransformer(model_save_path)
test_evaluator = EmbeddingSimilarityEvaluator.from_input_examples(test_samples, name='sts-test')
test_evaluator(model, output_path=model_save_path)
```

```
0.8289620954460468
```

Figure 16:

8. Applications

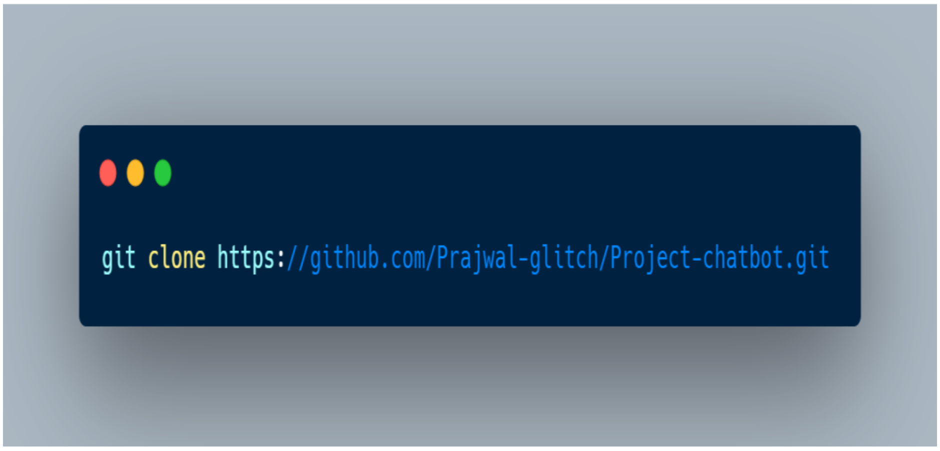
8. Applications

Chatbots - In recent times, chatbots are used extensively by people for various reasons, to get assistance in solving a particular problem (and getting it done fast) or to search information from the internet and much more. But these chatbots require a lot of training data. This data is needed to be in a specific format like yaml/json. This system can help resolve this issue. It requires minimal training data and will make sure that the answers will not go out of context. Also, the replies from the bots would be faster.

9. Installation Guide and User Manual

9. Installation Guide and User Manual

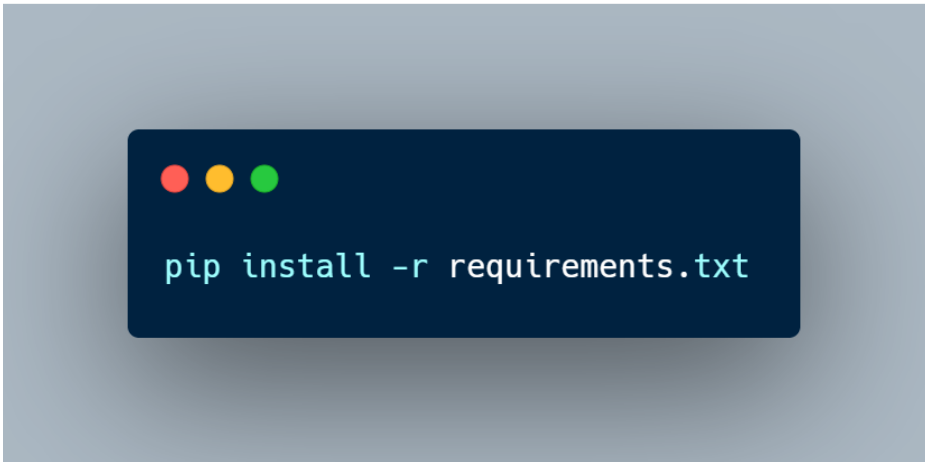
1. Download the code repository from github - https://github.com/Prajwal-glitch/Project-chatbot or use terminal to execute the following command



```
git clone https://github.com/Prajwal-glitch/Project-chatbot.git
```

Figure 17:

2. Once the repository is downloaded , you need to meet all the requirements.To install all dependencies execute the following command.

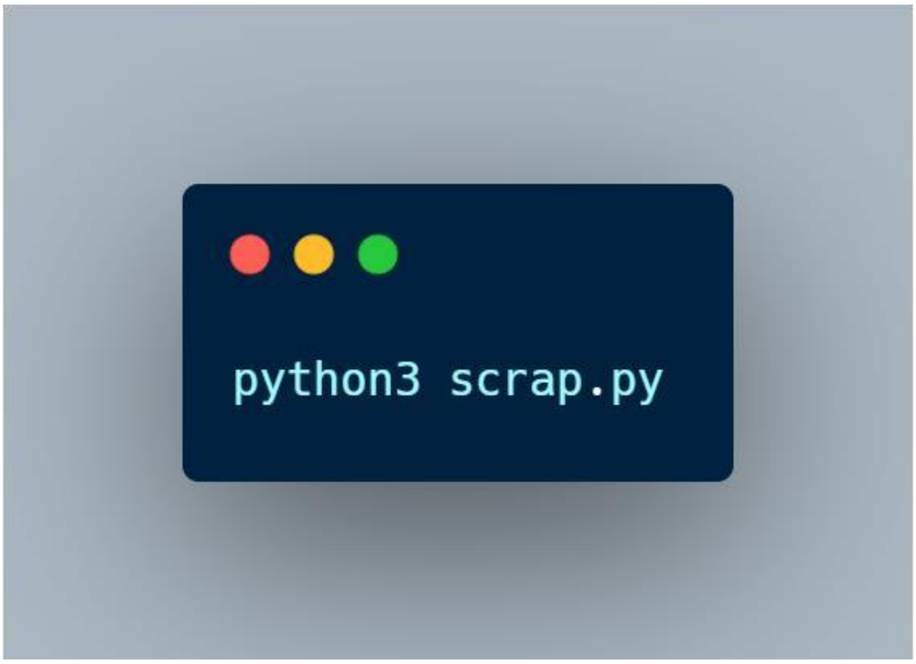


```
pip install -r requirements.txt
```

Figure 18:

3. Now run the scarp.py to create/update the question-answer database .

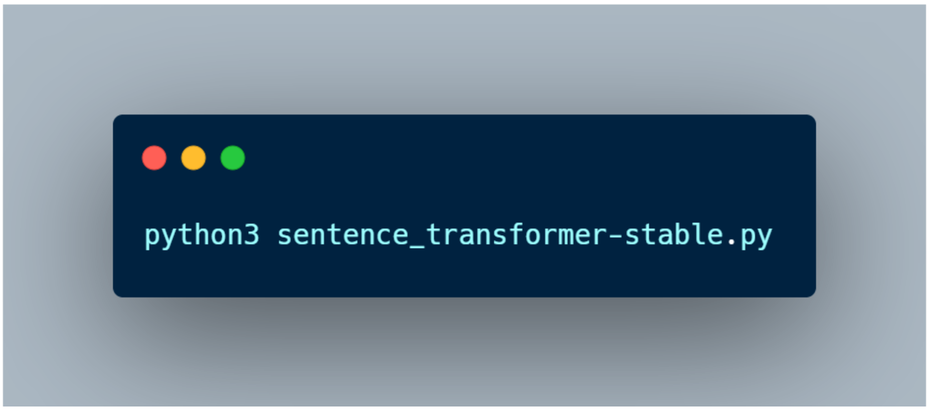
4. Now run the application by using following command



```
python3 scrap.py
```

Figure 19:

5. Now the application is running on the local server at https://127.0.0.1:5000/.Asimple GUI will be available to enter your queries and get responses to them.

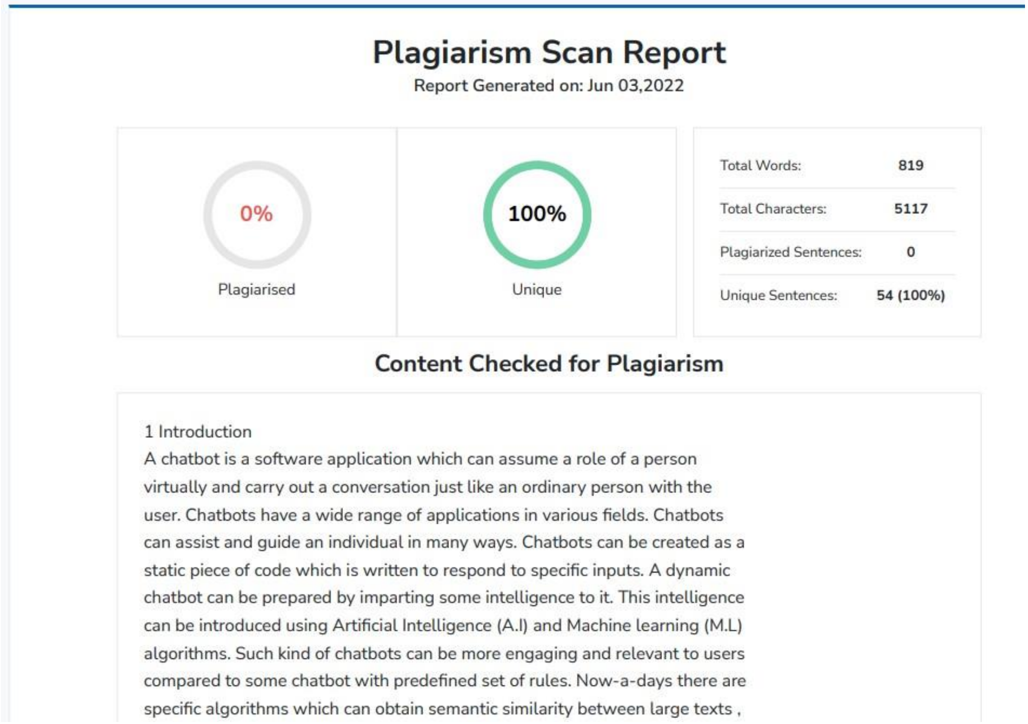


```
python3 sentence_transformer-stable.py
```

Figure 20:

10. Plagiarism Report

10. Plagiarism Report



Plagiarism Scan Report

Report Generated on: Jun 03,2022

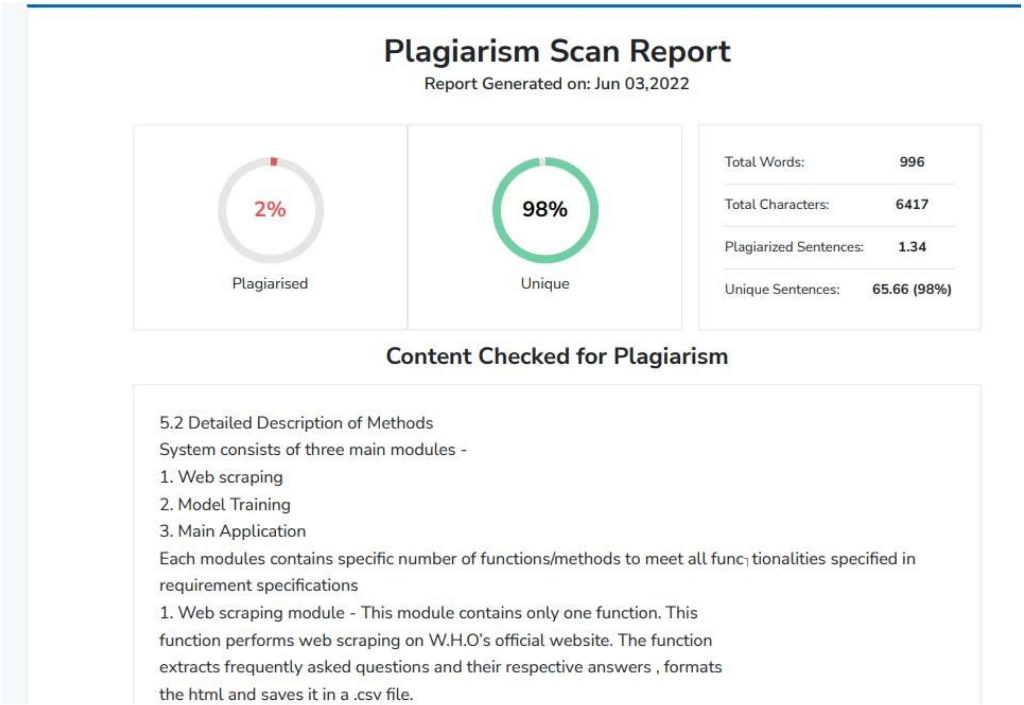
0% Plagiarised

100% Unique

Total Words:	819
Total Characters:	5117
Plagiarized Sentences:	0
Unique Sentences:	54 (100%)

Content Checked for Plagiarism

1 Introduction
A chatbot is a software application which can assume a role of a person virtually and carry out a conversation just like an ordinary person with the user. Chatbots have a wide range of applications in various fields. Chatbots can assist and guide an individual in many ways. Chatbots can be created as a static piece of code which is written to respond to specific inputs. A dynamic chatbot can be prepared by imparting some intelligence to it. This intelligence can be introduced using Artificial Intelligence (A.I) and Machine learning (M.L) algorithms. Such kind of chatbots can be more engaging and relevant to users compared to some chatbot with predefined set of rules. Now-a-days there are specific algorithms which can obtain semantic similarity between large texts ,



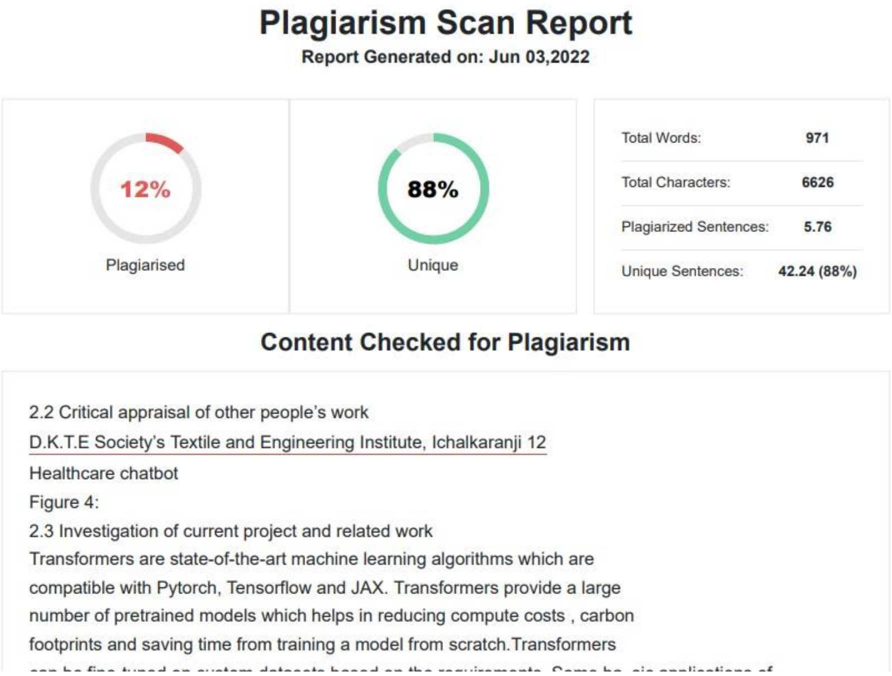
Plagiarism Scan Report

Report Generated on: Jun 03,2022

Total Words:	996
Total Characters:	6417
Plagiarized Sentences:	1.34
Unique Sentences:	65.66 (98%)

Content Checked for Plagiarism

5.2 Detailed Description of Methods
System consists of three main modules -
1. Web scraping
2. Model Training
3. Main Application
Each modules contains specific number of functions/methods to meet all func┐tionalities specified in requirement specifications
1. Web scraping module - This module contains only one function. This function performs web scraping on W.H.O's official website. The function extracts frequently asked questions and their respective answers , formats the html and saves it in a .csv file.



Plagiarism Scan Report

Report Generated on: Jun 03,2022

Total Words:	971
Total Characters:	6626
Plagiarized Sentences:	5.76
Unique Sentences:	42.24 (88%)

Content Checked for Plagiarism

2.2 Critical appraisal of other people's work
D.K.T.E Society's Textile and Engineering Institute, Ichalkaranji 12
Healthcare chatbot
Figure 4:
2.3 Investigation of current project and related work
Transformers are state-of-the-art machine learning algorithms which are
compatible with Pytorch, Tensorflow and JAX. Transformers provide a large
number of pretrained models which helps in reducing compute costs , carbon
footprints and saving time from training a model from scratch.Transformers
can be fine-tuned on custom datasets based on the requirements. Some ba sic applications of

11. Ethics

11. Ethics

Creative Commons License

Attribution-Share Alike 4.0 International (CC BY

SA 4.0) This is human-readable summary of (and not a

substitute for) the license.**You are free to:**

Share- copy and redistribute the material in any medium or format.

Adapt- remix, transform, and build upon the material for any purpose, even commercially.The licensor cannot revoke these freedoms as long as you follow the license terms.

Using the following terms:

Attribution you must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any responsible manner, but not in any way that suggests the licensor endorses you or your use.

Share Alike - you remix, transform, and build upon the material for any purpose, you mustdistribute your contributions under the same license as the original.

No additional restrictions- You must apply legal terms or technological measures thatlegally restrict others from doing anything the license permits.

Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an application exception or limitation. No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit hoe you use the material.

Ethical Practices for CSE Students:

As Computer Sc. & Engineering student, I believe it is unethical to:

- Take credit for someone else's work
- Hire someone to write an assignments
- Purchase or submit a research or term paper from the internet to a class as one's ownwork
- Cheat on a graded assignment
- Cheat on an exam
- Plagiarize other people's work without citing or referencing the work
- Add the name of a non-contributing person as an author in a project/research study
- Copy and paste material found on the Internet for an assignment without acknowledging the authorsof the material
- Deliberately provide inaccurate references for a project or research study
- Knowingly permit student work done by one student to be submitted by another student
- Surf the internet for personal interest and non-class related purposes during classes
- Make a copy of software for personal or commercial use
- Make a copy of software for a friend Loan CDs of software to friends
- Download pirated software from the internet
- Distribute pirated software from the internet
- Buy software with a single user license and then install it on multiple computers
- Share a pirated copy of software
- Install a pirated copy of software

12. References

12. References

[1] Mohammed Javed, P. Nagabhushan, B.B. Chaudhari, “A Direct Approach for Word and Character Segmentation in Run-Length Compressed Documents with an Application to Word Spotting”, 13th International Conference on Document Analysis and Recognition (ICDAR), 2015.

[2] Naeun Lee, Kirak Kim, Taeseon Yoon, “Implementation of Robot Journalism by Programming Custom bot using Tokenization and Custom Tagging”, 2017

[3] Jerome r. Bellagarda, “Parts-Of-Speech tagging by Latent Analogy”, IEEE Journal of Selected Topics in Signal Processing, Vol. 4, No. 6, 2010.

[4] Bo Chen, Donghong Ji, “Chinese Semantic Parsing based on Dependency Graph and Feature Structure”,
International Conference on Electronic and Mechanical Engineering and Information Technology, 2011

[5] Sachin S. Gavankar, Sudhirkumar D. Sawarkar, “Eager Decision Tree”, 2nd International Conference for Convergence in Technology (I2CT), 2017

[6] Sijun Qin, Jia Song, Pengzhou Zang, Yue Tan, “Feature Selection for Text Classification Based on Parts- Of-Speech Filter and Synonym Merge”, 12th International Conference on Fuzzy Systems and Knowledge Discover (FSKD), 2015

[7] https://uccs.edu/Documents/tboult/srs.doc

[8] https://hellotars.com/chatbot-templates/healthcare/

[9]https://towardsdatascience.com/how-we-created-an-open-source-covid-19- chatbot-c5c900b382dfbf0e

[10] https://www.sbert.net/

[11] https://huggingface.co/

[12] https://jalammar.github.io/illustrated-transformer/