Emergency Assistance Chatbot

Team member: Nihal Singh, Preetam Soni, and Chunwang Yuan
Advisor: Vahid Behzadan

Abstract

The sudden injury and illness do bring deadly consequences and exposed the limitations of the current process at the time of emergency staff to reach the scene, and the personnel at the scene lacked first aid skills.
In this work, let's develop an intelligent general chat robot to help the victims to help avoid worsening the condition of the victim and maintain its physical integrity, and opportunities for survival. Therefore, even people without first aid skills can help the victims survive through the implementation of the execution of emergency support. In addition, because it relies on emergency medical processing data and experience, it is trustworthy.
The proposed chatbot is based on the two methods. First, we are building our model from scratch using TensorFlow deep neural networks. And another method is fine-tuning existing pre-trained models like Biobert.
We provide our code here: https://github.com/ChunwangYuan/Emergency-assistance-chatbot

KEY WORDS: Emergency Chatbot, DNN, Fine-tuning, pre-trained, Biobert.

1 Introduction

For the sudden injury and illness, proper assistance in the early few minutes of an accident can help save a life. Also, for every minor problem, it is not necessary to see a medical expert. A proper and efficient working chatbot is very much in demand in the medical sector.
In this case study, we want to automate some of these queries. If a patient asks a question, our chatbot that has been trained on the question-answering model and dataset can give a proper answer. If the patient doesn't get his or her answer, we forward him to an actual doctor.
In this way, we can ensure the emergency rescue time for patients. And because they are filtered out at every stage of the chatbot, the number of patients calling emergency services is drastically reduced. At the same time, enable the patient to take quick action to solve their problems.

2 Methodology

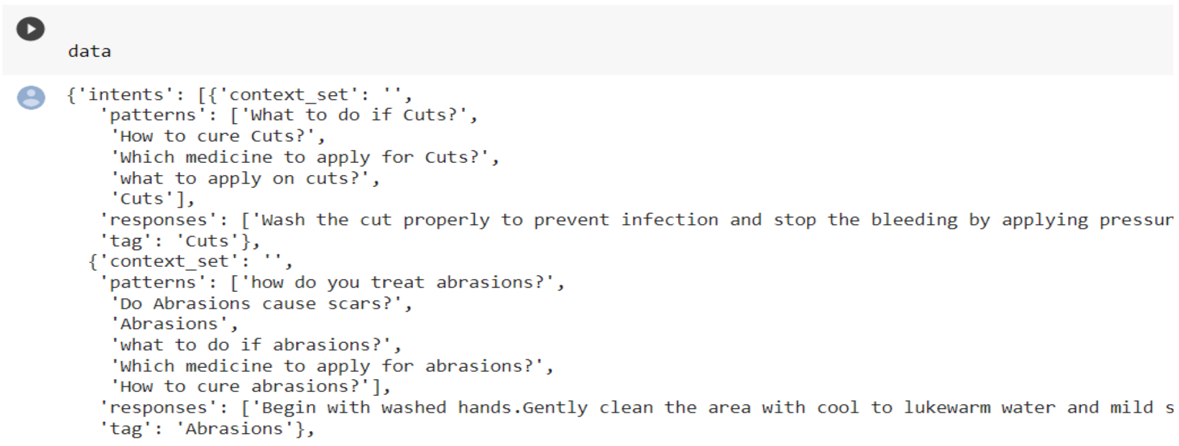
In this project we are using two methods to build the chatbot, first, we are building our model from scratch using TensorFlow deep neural networks and another method is fine-tuning existing pre-trained models like Biobert.

2.1 Method-1

We searched for relevant articles in literature databases (IEEE, ACM, Springer, ScienceDirect, Embase, MEDLINE, PsycINFO, and Google Scholar) [1]. After extensive research, we decided that we would develop the backend of the chatbot using two methods. The first method is based on the concept of the bag of the word [2]. After understanding the method, we researched the dataset. We found two datasets. The dataset for method 1 and method 2 is different. The dataset was loaded into google drive and we imported all the libraries used in this project like nltk and TensorFlow [3].

i) Load data:

Our dataset in JSON format, it is kind of python dictionary format. So, the access of this kind of data is bit tricky because each data set is paired key and value format. So, it was very important to be aware of our key and value pair when we were trying to access the data.

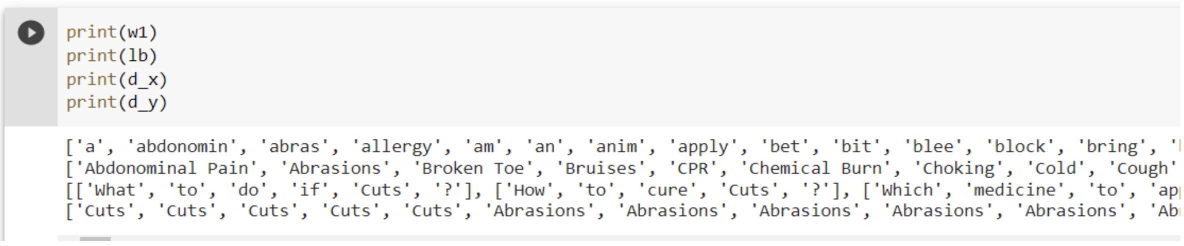


```
data

{'intents': [{'context_set': '',
   'patterns': ['What to do if Cuts?',
    'How to cure Cuts?',
    'Which medicine to apply for Cuts?',
    'what to apply on cuts?',
    'Cuts'],
   'responses': ['Wash the cut properly to prevent infection and stop the bleeding by applying pressur
   'tag': 'Cuts'},
  {'context_set': '',
   'patterns': ['how do you treat abrasions?',
    'Do Abrasions cause scars?',
    'Abrasions',
    'what to do if abrasions?',
    'Which medicine to apply for abrasions?',
    'How to cure abrasions?'],
   'responses': ['Begin with washed hands.Gently clean the area with cool to lukewarm water and mild s
   'tag': 'Abrasions'},
```

i) Data preparations:

Once the data and library are imported, the next step was pre-processing the data.

Data preparations involve extracting a list of classes from input data. We created a stemmatized and lemmatized [4] list of vocabulary from input data.

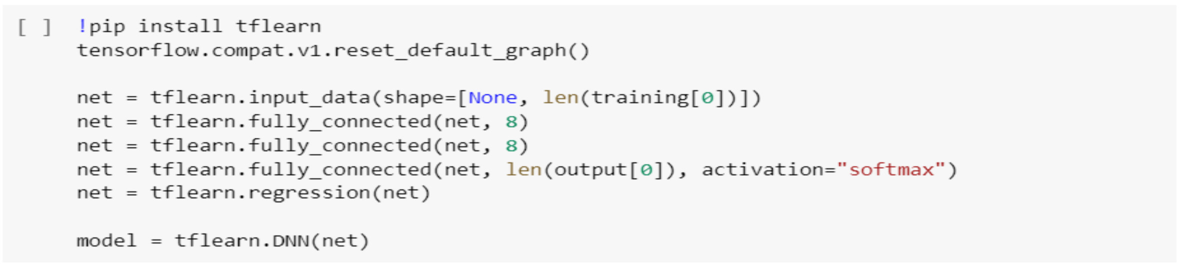


```
print(w1)
print(lb)
print(d_x)
print(d_y)

['a', 'abdonomin', 'abras', 'allergy', 'am', 'an', 'anim', 'apply', 'bet', 'bit', 'blee', 'block', 'bring', '
['Abdonominal Pain', 'Abrasions', 'Broken Toe', 'Bruises', 'CPR', 'Chemical Burn', 'Choking', 'Cold', 'Cough'
[['What', 'to', 'do', 'if', 'Cuts', '?'], ['How', 'to', 'cure', 'Cuts', '?'], ['Which', 'medicine', 'to', 'ap
['Cuts', 'Cuts', 'Cuts', 'Cuts', 'Cuts', 'Abrasions', 'Abrasions', 'Abrasions', 'Abrasions', 'Abrasions', 'Ab
```

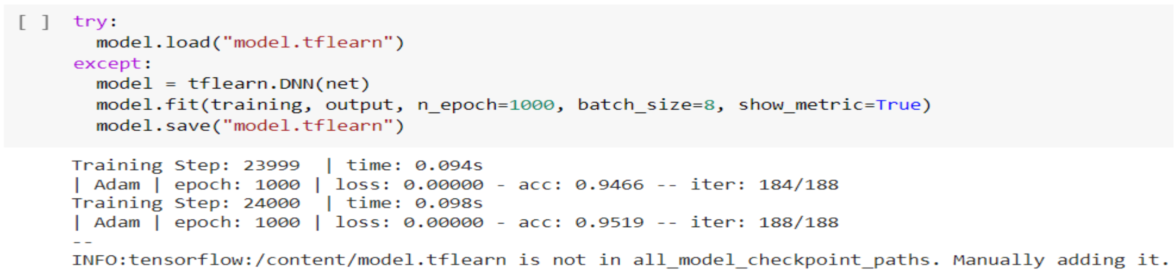
After that, we created a bag of words for training [5] and split it into a training and test that is used during model training and evaluation but the performance got was not very good as the data was very small [5]. So, we used the complete dataset for training considering the size of our data set [6].

ii) Model Creation and Training:

We used a very simple neural network [7]. Our neural networks have an input layer, two linear fully connected layers and the output layer has a SoftMax activation function [8]. The network is trained over 1000 epochs [9].



```
[ ]  !pip install tflearn
     tensorflow.compat.v1.reset_default_graph()

     net = tflearn.input_data(shape=[None, len(training[0])])
     net = tflearn.fully_connected(net, 8)
     net = tflearn.fully_connected(net, 8)
     net = tflearn.fully_connected(net, len(output[0]), activation="softmax")
     net = tflearn.regression(net)

     model = tflearn.DNN(net)
```



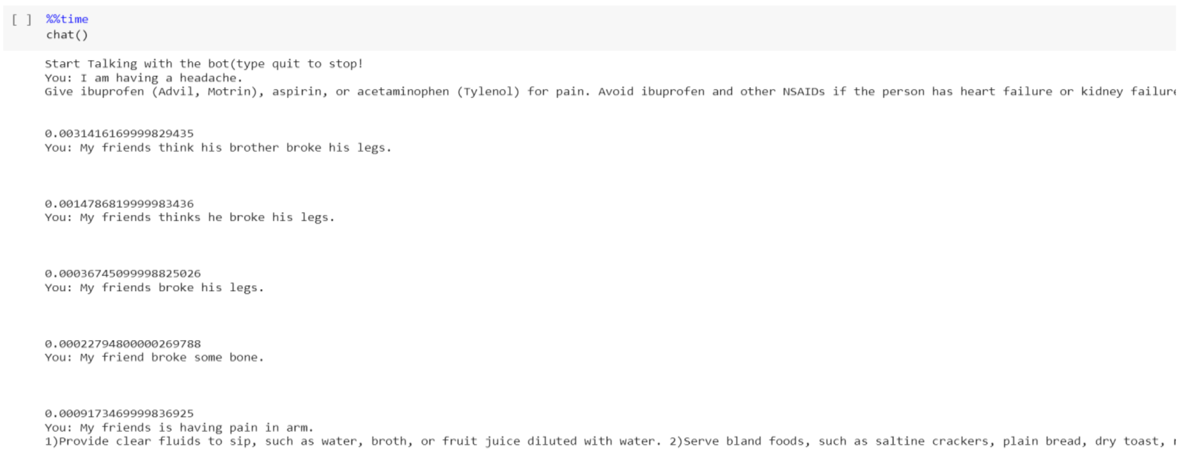
```
[ ]  try:
       model.load("model.tflearn")
     except:
       model = tflearn.DNN(net)
       model.fit(training, output, n_epoch=1000, batch_size=8, show_metric=True)
       model.save("model.tflearn")

     Training Step: 23999  | time: 0.094s
     | Adam | epoch: 1000 | loss: 0.00000 - acc: 0.9466 -- iter: 184/188
     Training Step: 24000  | time: 0.098s
     | Adam | epoch: 1000 | loss: 0.00000 - acc: 0.9519 -- iter: 188/188
     --
     INFO:tensorflow:/content/model.tflearn is not in all_model_checkpoint_paths. Manually adding it.
```

Finally created a user-defined function that takes user input pre-processes it and predicts the respective tag using our trained model and collects the information associated with the tag and displays the result to the user [10].

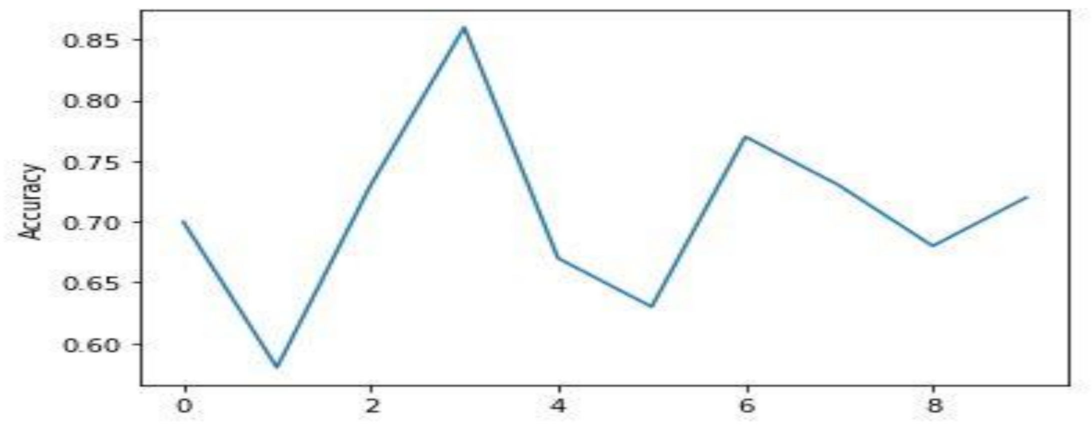
iii) Model Test and Results:

The model is based on a bag of words and trained on a neural network to predict the right tag from the bag of words so that the correct response can be provided to the user.

The output performed by the trained DNN model is showed at the below:



```
[ ]  %%time
     chat()

     Start Talking with the bot(type quit to stop!
     You: I am having a headache.
     Give ibuprofen (Advil, Motrin), aspirin, or acetaminophen (Tylenol) for pain. Avoid ibuprofen and other NSAIDs if the person has heart failure or kidney failur


     0.0031416169999829435
     You: My friends think his brother broke his legs.


     0.0014786819999983436
     You: My friends thinks he broke his legs.


     0.00036745099998825026
     You: My friends broke his legs.


     0.00022794800000269788
     You: My friend broke some bone.


     0.0009173469999836925
     You: My friends is having pain in arm.
     1)Provide clear fluids to sip, such as water, broth, or fruit juice diluted with water. 2)Serve bland foods, such as saltine crackers, plain bread, dry toast,
```



The model has a swift response and a good accuracy for now. But there are a lot works to do in the future to improve the model.

2.2 Method-2

In this method, we aim to automate some of these query resolutions with automation. This way when the user asks a question, the chatbot itself that has been trained on such question-answer pairs can try to come up with an answer. If the user doesn't get his or her answer, then we may redirect him to an actual doctor.

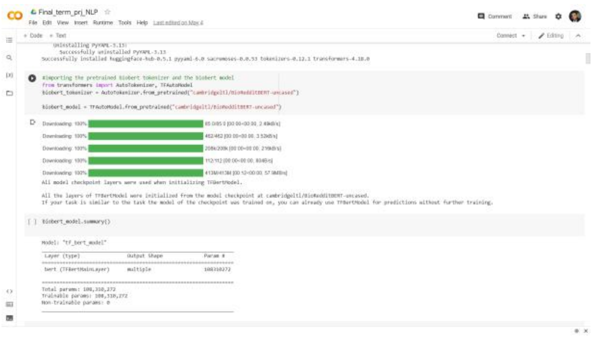
There are main components to solve the problem which might include many sub-components. Since at first, we are trying to generate an answer to the patient's query we can pose this problem as a seq2seq task trained in supervised fashion from previously answered question-answer pairs.

Dataset Analysis: we have already gathered data and described it. we will be using same data for this method as well along with this new data. We use the medical question answer dataset prepared by Lasse Regin Nelson. This dataset is uploaded in his GitHub https://github.com/LasseRegin/medical-question-answer-data. He has gathered such question answer pairs from prominent medical websites such as eHealth Forum, iCliniq, Question Doctors, WebMD where real doctors have provided public answers to

the questions asked by patients. We thank the above websites and Lasse Regin for helping us with the dataset and enabling us to help the medical community further.
There are different components in this task which require different performance metrics. For the task of seq2seq generation, instead of relying on bleu score for measuring the efficiency of our model, we rather evaluate it by the manual interpretation of the quality of answers generated. The reason for not using bleu score is, even though it is generally used for evaluating seq2seq tasks, it is still not highly reliable for us to judge our model on this bleu score, at least for our biomedical task, as for some sentences it may give a very good bleu score and for some others it might give very low bleu scores. To better judge the model, the answers generated by the model should be open sourced to be judged by medical experts, only then we can come up with a correct evaluation of the model. For the task of semantic search although we do not have a labelled dataset indicating which questions are like calculate the overall metric, we use cosine similarity to evaluate the embeddings of question and answer returned by the model. This link refers to the BIOBERT: a pre-trained biomedical language representation model for biomedical text mining proposed by Jinhyuk Lee et.al.[1].This link refers to the "Language Models are Unsupervised Multi task Learners" paper proposed by Alec Radford et.al.[12].The above link refers to the GitHub page of Doc Product: Medical Q&A with deep learning models[13].This encoding of question and answers is done by using the same pre-trained BioBert model that encodes both the given question and the given answer of the training dataset separately, whose output is passed to two separate fully connected neural networks to get final question and answer representations. These FCNN are trained using the dataset whose positive samples are created by taking question and answer from the same pair and negative samples are created by taking question and answer from different pairs. Our overall architecture is inspired from our reference DocProduct, from which we use the idea of using pre-trained BioBert model to extract embeddings for previously answered question and answer pairs and use these embeddings to do a semantic search on the existing question-answer pairs. We compared the cosine similarities returned by the model for the validation dataset for both correctly classified negative and positive points, as well as the cosine similarities returned by the model for misclassified points that are originally negative and positive points. Please have a look at the below plots for the same. Now given that we trained the model, let us look at how we extract the question-and-answer embeddings from the mode. We pass the question and answers to the fine-tuned BioBert model and pass it to both the residual blocks of question and answer and then we store both these embeddings along with the question and answers to be used for semantic search.
We generated answers for 5 questions each from both train and validation dataset using the model that gave least train loss. Although the bleu score is high for some of the answers and low for some other, overall, we were able to generate decent looking answers for most of the questions. Please have a looks at both the training and validation answers that have been generated. Note: The first line is the entire context, the second line is the actual answer, and the third line is the generated answer in each paragraph. In the future, we can experiment on how a transformer model can work on solving this problem directly as a seq2seq problem. Even if transformer model can directly generate answers to the given question, we still need BioBert to extract similar patient question-answer pairs. We can also work on gathering more biomedical data so that the model doesn't overfit to the given data and can generate more appropriate answers. As we tried to implement this process with online reference, we had faced certain issues which are described below:

i) Dataset Preparation:

We installed transformers and loaded the datasets.



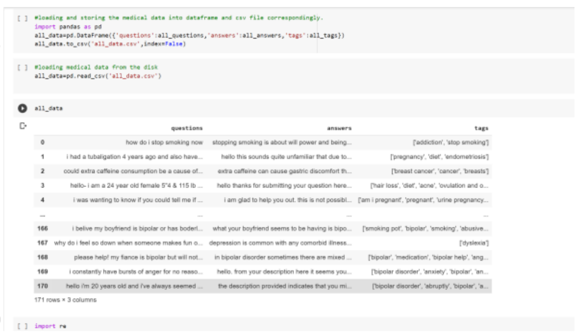
```
    Uninstalling PyYAML-3.13:
      Successfully uninstalled PyYAML-3.13
Successfully installed huggingface-hub-0.5.1 pyyaml-6.0 sacremoses-0.0.53 tokenizers-0.12.1 transformers-4.18.0
```

```
#importing the pretrained biobert tokenizer and the biobert model
from transformers import AutoTokenizer, TFAutoModel
biobert_tokenizer = AutoTokenizer.from_pretrained("cambridgeltl/BioRedditBERT-uncased")

biobert_model = TFAutoModel.from_pretrained("cambridgeltl/BioRedditBERT-uncased")
```

```
Downloading: 100%  85.0/85.0 [00:00<00:00, 2.40kB/s]
Downloading: 100%  462/462 [00:00<00:00, 3.53kB/s]
Downloading: 100%  208k/208k [00:00<00:00, 216kB/s]
Downloading: 100%  112/112 [00:00<00:00, 804B/s]
Downloading: 100%  413M/413M [00:12<00:00, 57.9MB/s]
All model checkpoint layers were used when initializing TFBertModel.

All the layers of TFBertModel were initialized from the model checkpoint at cambridgeltl/BioRedditBERT-uncased.
If your task is similar to the task the model of the checkpoint was trained on, you can already use TFBertModel for predictions without further training.
```

```
biobert_model.summary()
```

```
Model: "tf_bert_model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 bert (TFBertMainLayer)      multiple                  108310272

=================================================================
Total params: 108,310,272
Trainable params: 108,310,272
Non-trainable params: 0
_________________________________________________________________
```



```
#loading and storing the medical data into dataframe and csv file correspondingly.
import pandas as pd
all_data=pd.DataFrame({'questions':all_questions,'answers':all_answers,'tags':all_tags})
all_data.to_csv('all_data.csv',index=False)
```

```
#loading medical data from the disk
all_data=pd.read_csv('all_data.csv')
```

```
all_data
```

	questions	answers	tags
0	how do i stop smoking now	stopping smoking is about will power and being...	['addiction', 'stop smoking']
1	i had a tubaligation 4 years ago and also have...	hello this sounds quite unfamiliar that due to...	['pregnancy', 'diet', 'endometriosis']
2	could extra caffeine consumption be a cause of...	extra caffeine can cause gastric discomfort th...	['breast cancer', 'cancer', 'breasts']
3	hello- i am a 24 year old female 5"4 & 115 lb ...	hello thanks for submitting your question here...	['hair loss', 'diet', 'acne', 'ovulation and o...
4	i was wanting to know if you could tell me if ...	i am glad to help you out. this is not possibl...	['am i pregnant', 'pregnant', 'urine pregnancy...
...	...	...	...
166	i belive my boyfriend is bipolar or has boderl...	what your boyfriend seems to be having is bipo...	['smoking pot', 'bipolar', 'smoking', 'abusive...
167	why do i feel so down when someone makes fun o...	depression is common with any comorbid illness...	['dyslexia']
168	please help! my fiance is bipolar but will not...	in bipolar disorder sometimes there are mixed ...	['bipolar', 'medication', 'bipolar help', 'ang...
169	i constantly have bursts of anger for no reaso...	hello. from your description here it seems you...	['bipolar disorder', 'anxiety', 'bipolar', 'an...
170	hello i'm 20 years old and i've always seemed ...	the description provided indicates that you mi...	['bipolar disorder', 'abruptly', 'bipolar', 'a...

171 rows × 3 columns

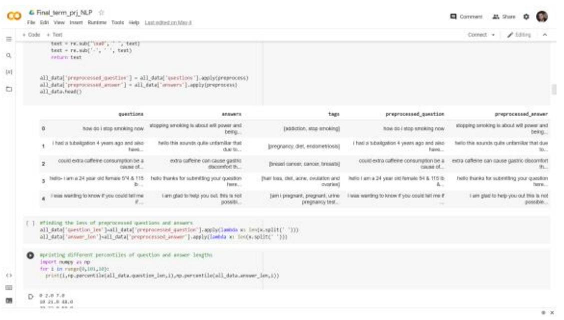
```
import re
```

After that we imported the pretrained biobert tokenizer and the biobert model. We defined function to extract question, answer, and tags from the json objects.

ii) Data Pre-processing:

We defined another function to preprocess tags. Preprocessing of the given question and answer data consisted of deconstructing some of the words such as "won't" to "will not", remove unnecessary symbols such as: !, etc. We also do not want to pad most of the question and answer with the padding constant.

After that we preprocessed all the questions and answers. We also extracted the positive and negative samples using tags.



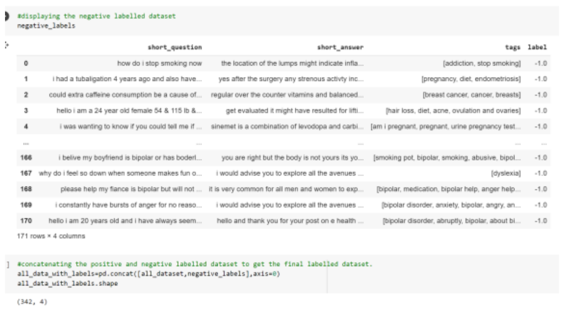
```
    text = re.sub("\xa0", ' ', text)
    text = re.sub('-', ' ', text)
    return text


all_data['preprocessed_question'] = all_data['questions'].apply(preprocess)
all_data['preprocessed_answer'] = all_data['answers'].apply(preprocess)
all_data.head()
```

	questions	answers	tags	preprocessed_question	preprocessed_answer
0	how do i stop smoking now	stopping smoking is about will power and being...	[addiction, stop smoking]	how do i stop smoking now	stopping smoking is about will power and being...
1	i had a tubaligation 4 years ago and also have...	hello this sounds quite unfamiliar that due to...	[pregnancy, diet, endometriosis]	i had a tubaligation 4 years ago and also have...	hello this sounds quite unfamiliar that due to...
2	could extra caffeine consumption be a cause of...	extra caffeine can cause gastric discomfort th...	[breast cancer, cancer, breasts]	could extra caffeine consumption be a cause of...	extra caffeine can cause gastric discomfort th...
3	hello- i am a 24 year old female 5'4 & 115 lb...	hello thanks for submitting your question here...	[hair loss, diet, acne, ovulation and ovaries]	hello i am a 24 year old female 54 & 115 lb &...	hello thanks for submitting your question here...
4	i was wanting to know if you could tell me if...	i am glad to help you out. this is not possibl...	[am i pregnant, pregnant, urine pregnancy test...	i was wanting to know if you could tell me if ...	i am glad to help you out this is not possible...

```
#finding the lens of preprocessed questions and answers
all_data['question_len']=all_data['preprocessed_question'].apply(lambda x: len(x.split(' ')))
all_data['answer_len']=all_data['preprocessed_answer'].apply(lambda x: len(x.split(' ')))
```

```
#printing different percentiles of question and answer lengths
import numpy as np
for i in range(0,101,10):
  print(i,np.percentile(all_data.question_len,i),np.percentile(all_data.answer_len,i))
```

```
0 2.0 7.0
10 21.0 48.0
```



```
#displaying the negative labelled dataset
negative_labels
```

	short_question	short_answer	tags	label
0	how do i stop smoking now	the location of the lumps might indicate infla...	[addiction, stop smoking]	-1.0
1	i had a tubaligation 4 years ago and also have...	yes after the surgery any strenous activty inc...	[pregnancy, diet, endometriosis]	-1.0
2	could extra caffeine consumption be a cause of...	regular over the counter vitamins and balanced...	[breast cancer, cancer, breasts]	-1.0
3	hello i am a 24 year old female 54 & 115 lb &...	get evaluated it might have resulted for lifti...	[hair loss, diet, acne, ovulation and ovaries]	-1.0
4	i was wanting to know if you could tell me if ...	sinemet is a combination of levodopa and carbi...	[am i pregnant, pregnant, urine pregnancy test...	-1.0
...	...	...	...	...
166	i belive my boyfriend is bipolar or has boderl...	you are right but the body is not yours its yo...	[smoking pot, bipolar, smoking, abusive, bipol...	-1.0
167	why do i feel so down when someone makes fun o...	i would advise you to explore all the avenues ...	[dyslexia]	-1.0
168	please help my fiance is bipolar but will not ...	it is very common for all men and women to exp...	[bipolar, medication, bipolar help, anger help...	-1.0
169	i constantly have bursts of anger for no reaso...	i would advise you to explore all the avenues ...	[bipolar disorder, anxiety, bipolar, angry, an...	-1.0
170	hello i am 20 years old and i have always seem...	hello and thank you for your post on e health ...	[bipolar disorder, abruptly, bipolar, about bi...	-1.0

171 rows × 4 columns

```
#concatenating the positive and negative labelled dataset to get the final labelled dataset.
all_data_with_labels=pd.concat([all_dataset,negative_labels],axis=0)
all_data_with_labels.shape
```

```
(342, 4)
```

After that we concatenated the positive and negative labelled dataset to get the final labelled dataset which is split into train and validation dataset.

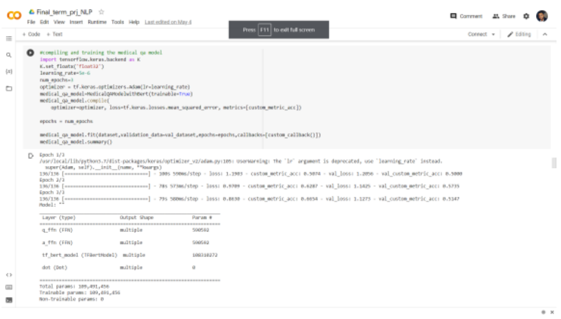
```
[ ] #concatenating the positive and negative labelled dataset to get the final labelled dataset.
    all_data_with_labels=pd.concat([all_dataset,negative_labels],axis=0)
    all_data_with_labels.shape

    (342, 4)
```

```
[ ] #splitting the data into train and validation
    from sklearn.model_selection import train_test_split
    train, validation = train_test_split(all_data_with_labels, test_size=0.2,random_state=42,shuffle=True,stratify=all_data_with_labels.label)
```

Now, we have our data ready let us see how we have defined the question extractor model, answer extractor model and how we fine-tuned the pre-trained hugging-face BioBert transformer to achieve the task of classifying whether a given question and answer relate to the same context.

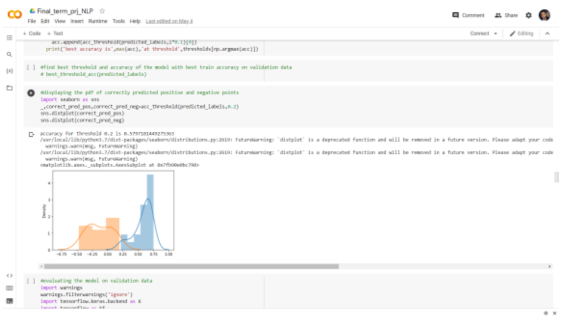
iii) Model Creation and Training:

We designed the feed forward neural network. We also created a custom-made medical Bert model.



```
#compiling and training the medical qa model
import tensorflow.keras.backend as K
K.set_floatx("float32")
learning_rate=5e-6
num_epochs=3
optimizer = tf.keras.optimizers.Adam(lr=learning_rate)
medical_qa_model=MedicalQAModelwithBert(trainable=True)
medical_qa_model.compile(
    optimizer=optimizer, loss=tf.keras.losses.mean_squared_error, metrics=[custom_metric_acc])

epochs = num_epochs

medical_qa_model.fit(dataset,validation_data=val_dataset,epochs=epochs,callbacks=[custom_callback()])
medical_qa_model.summary()
```

```
Epoch 1/3
/usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/adam.py:105: UserWarning: The `lr` argument is deprecated, use `learning_rate` instead.
  super(Adam, self).__init__(name, **kwargs)
136/136 [==============================] - 100s 590ms/step - loss: 1.1903 - custom_metric_acc: 0.5074 - val_loss: 1.2056 - val_custom_metric_acc: 0.5000
Epoch 2/3
136/136 [==============================] - 78s 573ms/step - loss: 0.9709 - custom_metric_acc: 0.6287 - val_loss: 1.1425 - val_custom_metric_acc: 0.5735
Epoch 3/3
136/136 [==============================] - 79s 580ms/step - loss: 0.8630 - custom_metric_acc: 0.6654 - val_loss: 1.1273 - val_custom_metric_acc: 0.5147
Model: ""
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 q_ffn (FFN)                 multiple                  590592

 a_ffn (FFN)                 multiple                  590592

 tf_bert_model (TFBertModel)  multiple                 108310272

 dot (Dot)                   multiple                  0

=================================================================
Total params: 109,491,456
Trainable params: 109,491,456
Non-trainable params: 0
```



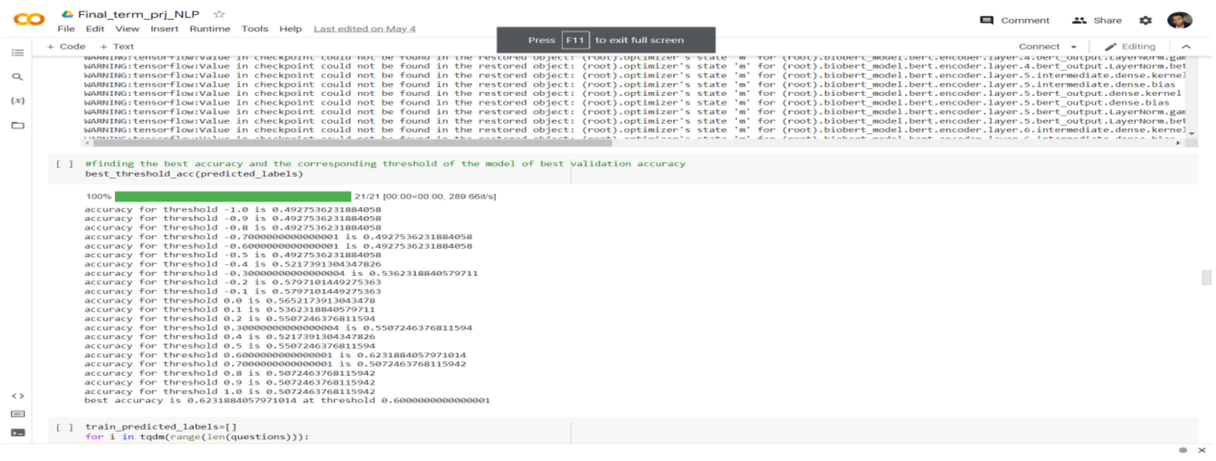
```
    acc.append(acc_threshold(predicted_labels,i*0.1)[0])
  print('best accuracy is',max(acc),'at threshold',thresholds[np.argmax(acc)])
```

```
#find best threshold and accuracy of the model with best train accuracy on validation data
# best_threshold_acc(predicted_labels)
```

```
#displaying the pdf of correctly predicted positive and negative points
import seaborn as sns
_,correct_pred_pos,correct_pred_neg=acc_threshold(predicted_labels,0.2)
sns.distplot(correct_pred_pos)
sns.distplot(correct_pred_neg)
```

```
accuracy for threshold 0.2 is 0.5797101449275363
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7fb90e0bc790>
```

```
#evaluating the model on validation data
import warnings
warnings.filterwarnings('ignore')
import tensorflow.keras.backend as K
```



```
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).biobert_model.bert.encoder.layer.4.bert_output.LayerNorm.bet
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).biobert_model.bert.encoder.layer.5.intermediate.dense.kernel
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).biobert_model.bert.encoder.layer.5.intermediate.dense.bias
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).biobert_model.bert.encoder.layer.5.bert_output.dense.kernel
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).biobert_model.bert.encoder.layer.5.bert_output.dense.bias
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).biobert_model.bert.encoder.layer.5.bert_output.LayerNorm.gam
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).biobert_model.bert.encoder.layer.5.bert_output.LayerNorm.bet
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm' for (root).biobert_model.bert.encoder.layer.6.intermediate.dense.kernel
```

```
#finding the best accuracy and the corresponding threshold of the model of best validation accuracy
best_threshold_acc(predicted_labels)
```

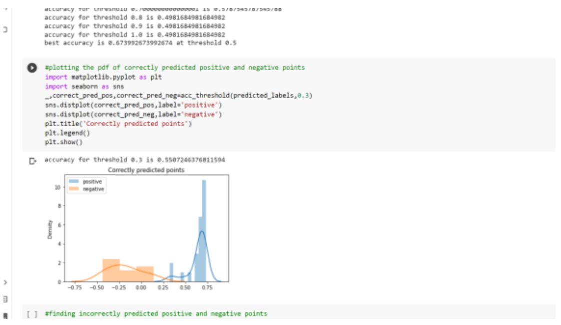
```
100% 21/21 [00:00<00:00, 289.66it/s]
accuracy for threshold -1.0 is 0.4927536231884058
accuracy for threshold -0.9 is 0.4927536231884058
accuracy for threshold -0.8 is 0.4927536231884058
accuracy for threshold -0.7000000000000001 is 0.4927536231884058
accuracy for threshold -0.6000000000000001 is 0.4927536231884058
accuracy for threshold -0.5 is 0.4927536231884058
accuracy for threshold -0.4 is 0.5217391304347826
accuracy for threshold -0.30000000000000004 is 0.5362318840579711
accuracy for threshold -0.2 is 0.5797101449275363
accuracy for threshold -0.1 is 0.5797101449275363
accuracy for threshold 0.0 is 0.5652173913043478
accuracy for threshold 0.1 is 0.5362318840579711
accuracy for threshold 0.2 is 0.5507246376811594
accuracy for threshold 0.30000000000000004 is 0.5507246376811594
accuracy for threshold 0.4 is 0.5217391304347826
accuracy for threshold 0.5 is 0.5507246376811594
accuracy for threshold 0.6000000000000001 is 0.6231884057971014
accuracy for threshold 0.7000000000000001 is 0.5072463768115942
accuracy for threshold 0.8 is 0.5072463768115942
accuracy for threshold 0.9 is 0.5072463768115942
accuracy for threshold 1.0 is 0.5072463768115942
best accuracy is 0.6231884057971014 at threshold 0.6000000000000001
```

```
train_predicted_labels=[]
for i in tqdm(range(len(questions))):
```

After that we created a custom metric accuracy. Finally compiling and training of the medical QA model.

iv) Model Evaluation on Validation Data:

We compared the cosine similarities returned by the model for the validation dataset for both correctly classified negative and positive points, as well as the cosine similarities returned by the model for misclassified points that are originally negative and positive points. Please have a look at the below plots for the same.

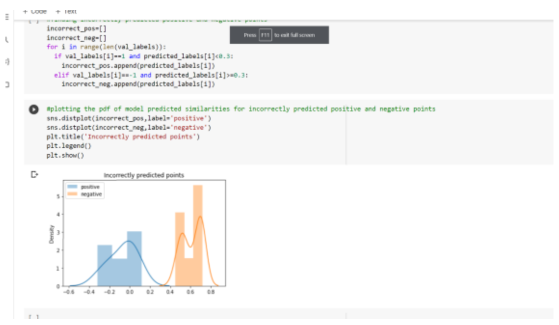


```
accuracy for threshold 0.8 is 0.4981684981684982
accuracy for threshold 0.9 is 0.4981684981684982
accuracy for threshold 1.0 is 0.4981684981684982
best accuracy is 0.673992673992674 at threshold 0.5
```

```
#plotting the pdf of correctly predicted positive and negative points
import matplotlib.pyplot as plt
import seaborn as sns
_,correct_pred_pos,correct_pred_neg=acc_threshold(predicted_labels,0.3)
sns.distplot(correct_pred_pos,label='positive')
sns.distplot(correct_pred_neg,label='negative')
plt.title('Correctly predicted points')
plt.legend()
plt.show()
```

```
accuracy for threshold 0.3 is 0.5507246376811594
```

```
#finding incorrectly predicted positive and negative points
```



```
incorrect_pos=[]
incorrect_neg=[]
for i in range(len(val_labels)):
  if val_labels[i]==1 and predicted_labels[i]<0.3:
    incorrect_pos.append(predicted_labels[i])
  elif val_labels[i]==-1 and predicted_labels[i]>=0.3:
    incorrect_neg.append(predicted_labels[i])
```

```
#plotting the pdf of model predicted similarities for incorrectly predicted positive and negative points
sns.distplot(incorrect_pos,label='positive')
sns.distplot(incorrect_neg,label='negative')
plt.title('Incorrectly predicted points')
plt.legend()
plt.show()
```

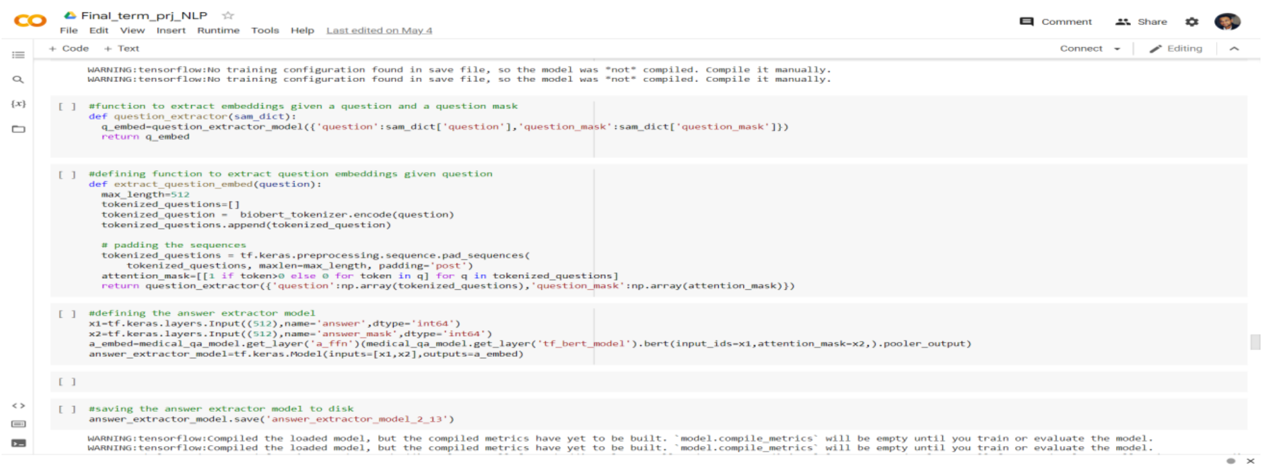
Correctly Classified Cosine Similarities Incorrectly Classified Cosine Similarities

From the plot above, we can observe that the correctly classified negative and positive points are well separated from each other. The threshold of 0.25 of model gave the best accuracy. We defined function to print the best accuracy and at the threshold which it is occurring.

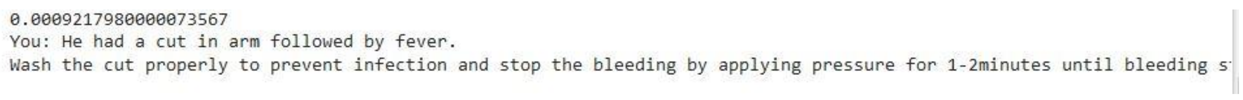
Now, we got the trained model, then we should extract the question and answer embeddings that both of them will be used for the semantic search.

v) Questions and Answers test results:

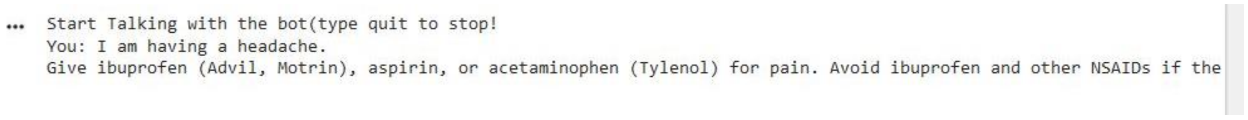
Please see the outcomes of our model at the below:


Final_term_prj_NLP
File Edit View Insert Runtime Tools Help Last edited on May 4
Comment
Share
+ Code + Text
Connect
Editing

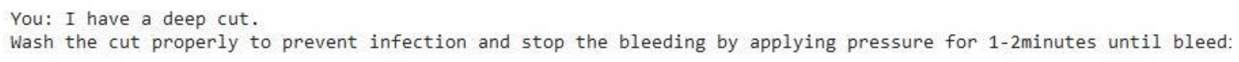
```
WARNING:tensorflow:No training configuration found in save file, so the model was *not* compiled. Compile it manually.
WARNING:tensorflow:No training configuration found in save file, so the model was *not* compiled. Compile it manually.

[ ] #function to extract embeddings given a question and a question mask
    def question_extractor(sam_dict):
      q_embed=question_extractor_model({'question':sam_dict['question'],'question_mask':sam_dict['question_mask']})
      return q_embed

[ ] #defining function to extract question embeddings given question
    def extract_question_embed(question):
      max_length=512
      tokenized_questions=[]
      tokenized_question =  biobert_tokenizer.encode(question)
      tokenized_questions.append(tokenized_question)

      # padding the sequences
      tokenized_questions = tf.keras.preprocessing.sequence.pad_sequences(
          tokenized_questions, maxlen=max_length, padding='post')
      attention_mask=[[1 if token>0 else 0 for token in q] for q in tokenized_questions]
      return question_extractor({'question':np.array(tokenized_questions),'question_mask':np.array(attention_mask)})

[ ] #defining the answer extractor model
    x1=tf.keras.layers.Input((512),name='answer',dtype='int64')
    x2=tf.keras.layers.Input((512),name='answer_mask',dtype='int64')
    a_embed=medical_qa_model.get_layer('a_ffn')(medical_qa_model.get_layer('tf_bert_model').bert(input_ids=x1,attention_mask=x2,).pooler_output)
    answer_extractor_model=tf.keras.Model(inputs=[x1,x2],outputs=a_embed)

[ ]

[ ] #saving the answer extractor model to disk
    answer_extractor_model.save('answer_extractor_model_2_13')

    WARNING:tensorflow:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or evaluate the model.
    WARNING:tensorflow:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or evaluate the model.
```



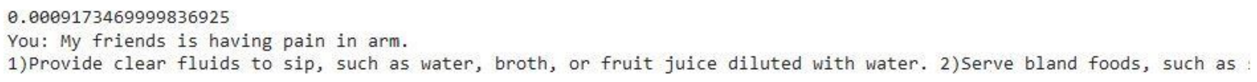
```
0.0009217980000073567
You: He had a cut in arm followed by fever.
Wash the cut properly to prevent infection and stop the bleeding by applying pressure for 1-2minutes until bleeding s
```



```
...  Start Talking with the bot(type quit to stop!
     You: I am having a headache.
     Give ibuprofen (Advil, Motrin), aspirin, or acetaminophen (Tylenol) for pain. Avoid ibuprofen and other NSAIDs if the
```



```
You: I have a deep cut.
Wash the cut properly to prevent infection and stop the bleeding by applying pressure for 1-2minutes until bleed:
```



```
0.0009173469999836925
You: My friends is having pain in arm.
1)Provide clear fluids to sip, such as water, broth, or fruit juice diluted with water. 2)Serve bland foods, such as
```

Research limitation

According to the survey, almost all chatbot development team have the medical expertise. However, we have no expert in this field. As a result, a lack of professional advice to provide us the expertise consulting may be one of the limitations of our project.

Conclusion

This time, we got a good performance of DNN model and pre-trained BioBert model. The DNN model has a decent performance of 75% accuracy. And we tried to use the pre-trained BioBert model which give us a good outcome. But we need to do more improvements to modify our model.
In the future, we can collect more related medical data and use more strategy to optimize our parameter so that the model doesn't overfitting to the training data and can get a better performance.
Find explanations: https://youtu.be/qPfvVoBay1E
Find more information: https://github.com/ChunwangYuan/Emergency-assistance-chatbot

Acknowledgement

All members from this team will express our sincere gratitude to the Prof. Behzadan for his valuable advice from beginning to the end of this project.

Reference:

[1] Understanding bag-of-words model: a statistical framework
[2] https://www.youtube.com/watch?v=7t1zZsVUs0
[3] https://www.ijcttjournal.org/2018/Volume60/number-1/IJCTT-V60P106.pdf
[4] https://www.jmir.org/2020/12/e19127/
[5] https://scholar.google.com/scholar?hl=en&as_sdt=0%2C7&q=chatbot&btnG=
[6] http://tflearn.org/
[7] https://www.youtube.com/watch?v=4h1H1F2WJ5Y
[8] https://www.techwithtim.net/tutorials/ai-chatbot/part-3/
[9] https://medium.com/@adeevamanal/building-an-ai-chatbot-to-answer-your-quarantine-questions-38d599278192
[10] https://app.grammarly.com/ddocs/1600973013
[11] https://arxiv.org/ftp/arxiv/papers/1901/1901.08746.pdf
[12] https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf
[13] https://github.com/ash3n/DocProduct#start-of-content
https://suniljammalamadaka.medium.com/medical-chatbot-using-bert-and-gpt2-62f0c973162f
https://marutitech.com/use-cases-of-natural-language-processing-in-healthcare/
https://www.youtube.com/results?search_query=biobert+question+answering+chatbot+for+medical+data
https://www.youtube.com/results?search_query=gpt2+question+answering
https://www.youtube.com/results?search_query=medical+chatbot+project
https://www.youtube.com/results?search_query=transfer+learning+for+computer+vision+tutorial+resnet+20+pytorch
https://www.youtube.com/watch?v=3XiJrn_8F9Q&t=92s
https://www.youtube.com/watch?v=3XiJrn_8F9Q&t=92s
https://www.youtube.com/watch?v=3XiJrn_8F9Q&t=92s
https://www.youtube.com/watch?v=3XiJrn_8F9Q&t=277s
https://www.youtube.com/watch?v=3XiJrn_8F9Q&t=277s
https://d4mucfpksywv.cloudfront.net/better-language-models/languagemodels.pdf%C2%A0
https://github.com/re-search/DocProduct#start-of-content