# Question Answering Chatbot System for Medical Questions

Zeinab Abdelmawla, *student* Ayman Abdeen, *student* and Micheal Waheeb, *student*

**Index Terms**—

## 1 Introduction

Difficulty in seeing a doctor, long queuing time, and inconvenience of making appointments have long been hurdles facing patients when they try to access primary care services. Adding also that the whole world is suffering until now from the global health cries Covid-19.

I wish you the best of success.

## 2 Problem statement

All of this increases the need of doctors to cover the growth of healthcare needs. so, governments and health care providers around the world are investing in new methods that facilitate more effective use of resources to meet demands. in the hope to better harness the power of digital medical data and information technology to deliver enhanced services.

## 3 Proposed solution

Artificial intelligence plays a crucial role in the advancement of information technology to improve healthcare service quality and efficiency. In particular, chatbots amount to one of the most popular AI technologies for this purpose. A chatbot is a software system that consists of an interactive interface with patients or medical practitioners to provide a range of knowledge extraction tasks and real-time, personalized feedback. Chatbot technologies have been rapidly developed, especially in the medical field. Many medical chatbot systems have been proposed over the years. Typical applications of chatbot include medical assistants that help patients to identify their symptoms, medical service front desks that direct the patient to suitable healthcare service departments, i.e., doctors, and so on. Our work aligns with the main themes of medical chatbot technology and aims to serve three main objectives: The first objective is to reduce waste on resources and time for users when accessing information with chatbot technologies. We aim to maximally help users to search for the necessary information with a human-like interface. The second objective is to provide more precise answers to ordinary users who have little domain knowledge. In other words, we hope that with AI technologies, the system can understand the meaning of the natural language and be able to reply with high-quality feedback accordingly. The third objective is to make it easier to manage and extend the features and databases. We want to design a system with a flexible and scalable structure to enable efficient management of the functionality and datasets.

## 4 Proposed solution framework

The diagram below of the new hierarchical BiLSTM Attention model we proposed is shown below in Figure 1. It is designed for semantic similarity comparison. The whole structure based on a Siamese LSTM framework . We apply one BiLSTM layer and one word attention layer into the Siamese framework. The bottom left, and the right sentences represent user input query and the question from the QA dataset. The two questions will be represented by using word embedding firstly and then using BiLSTM to form the whole sentence embedding based on the context. After that, each BiLSTM encoder will be multiplied by a word attention value, which can be assumed as a weight to highlight the key-point in a sentence. Context vector will be combined with attention to understanding the sentence representation uw

### 4.1 Model Structure

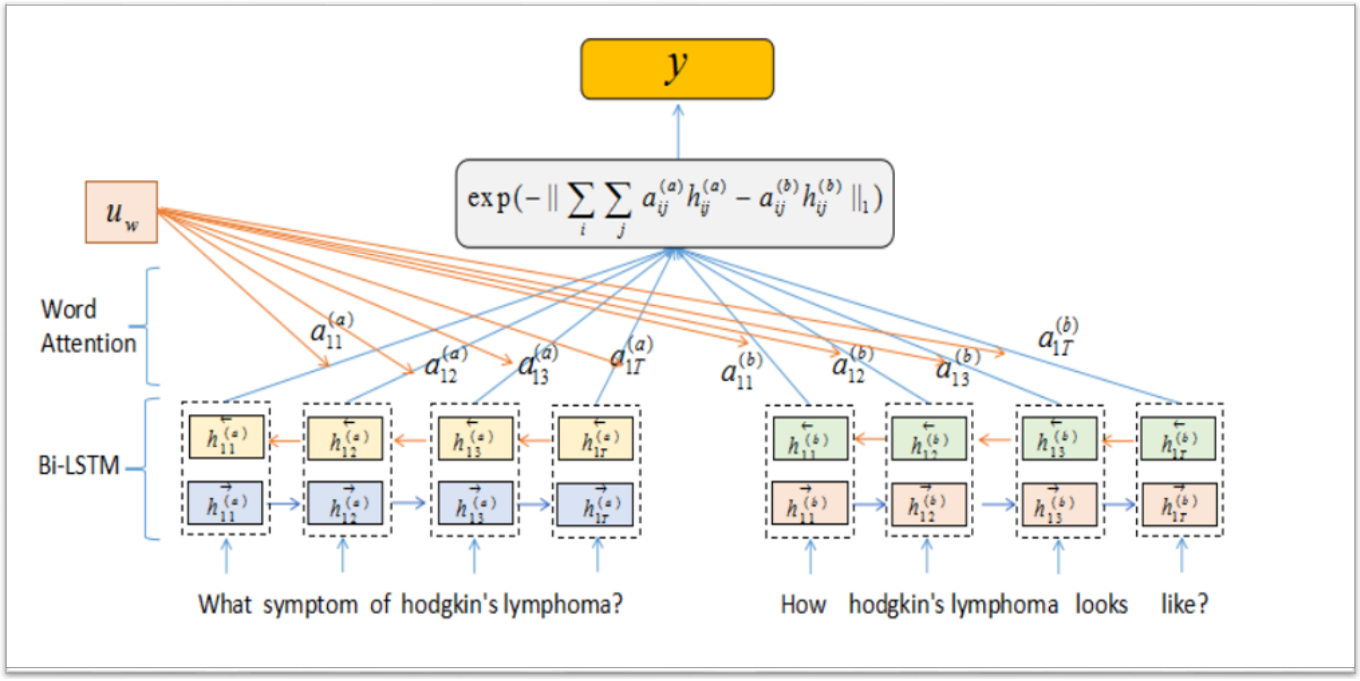


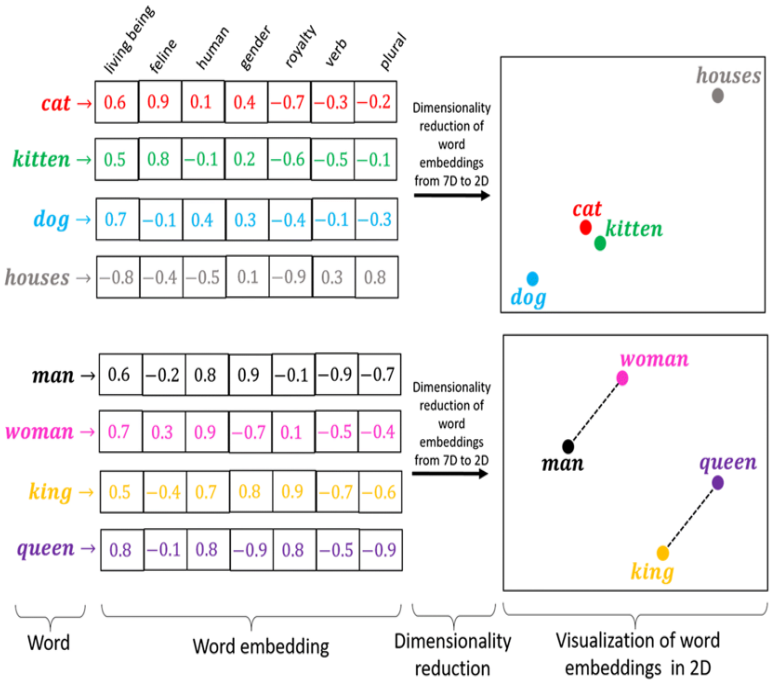

**Figure1:Hierarchical BiLSTM Attention Model**

### 4.2 Embedding equation

each word of the sentence will be embedded by using an embedding matrix W

$$x_{it} = W_e w_{it}, t \in [1, T]$$

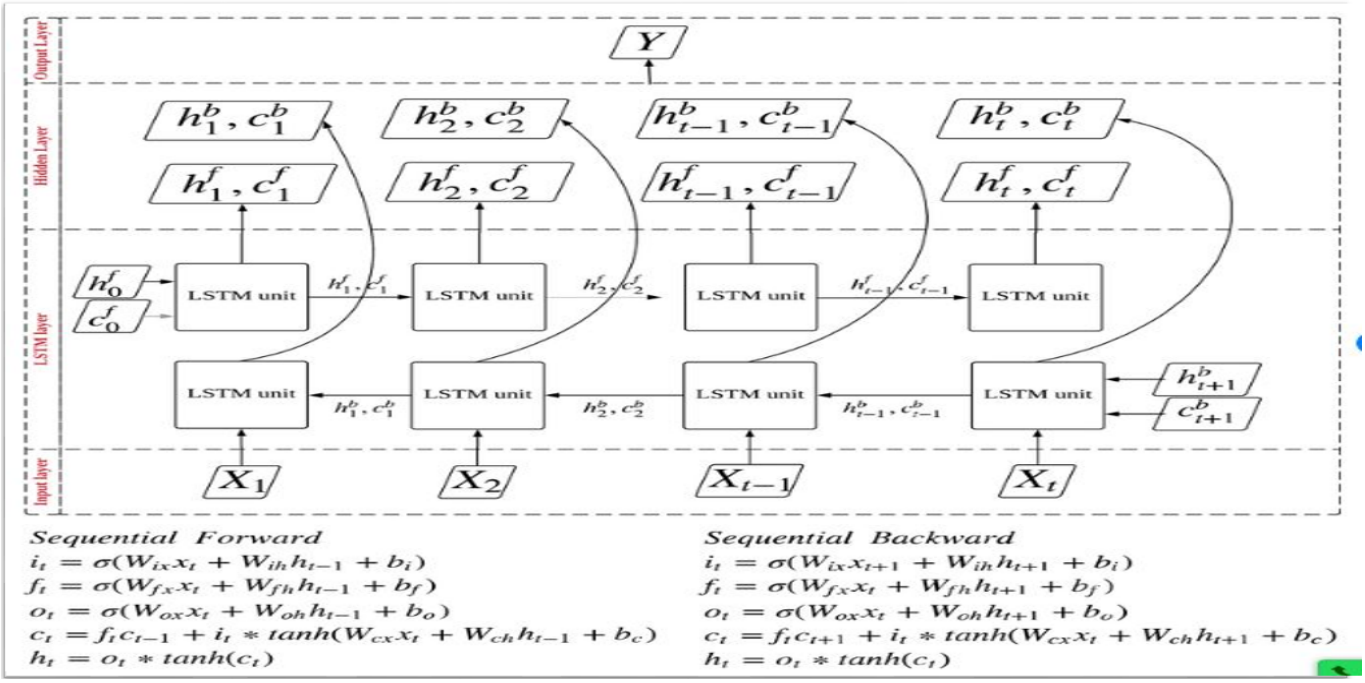
King - man + woman = queen



### 4.3 bilstm Structure equation

We use Bidirectional LSTM to capture both forward and reverse direction information of each word. The bidirectional LSTM contains forward LSTM forward and reverse LSTM backward.

$$\vec{h}_{it} = \vec{\text{LSTM}}(x_{it}), t \in [1, T]$$
$$\vec{h}_{ti} = \vec{\text{LSTM}}(x_{ti}), t \in [1, T]$$




### 4.4 Attention equation

We try to use Attention. Firstly, we feed the hit into the tanh function to get uit as a hidden representation of hit . Secondly, we calculate the importance of each word uit and get a normalized importance weight it by using a SoftMax function. Then, we calculate the sentence vector s1 as a weight sum of each word with its weight

$$u_{it} = \tanh(W_w h_{it} + b_w)$$
$$\alpha_{it} = \frac{\exp(u_{it})}{\sum_t \exp(u_{it})}$$
$$s_i = \sum_t \alpha_{it} h_{it}$$

### 4.5 Similarity function

The formula is based on Manhattan distance. From this formula, the representation from two sentences can represented

$$f\left(s_i^{(a)}, s_j^{(b)}\right) = \exp\left(-\left\|\sum_i \sum_j a_{ij}^{(a)} h_{ij}^{(a)} - a_{ij}^{(b)} h_{ij}^{(b)}\right\|_1\right) \in [0, 1]$$

# 5 EXPERIMENTS

## 5.1 Dataset

Train and Test Dataset. , we tried to train model on full Quora medical dataset as a part of our novility but not give us good results. so, we worked on selective Quora research medical dataset and was balanced.
The Quora duplicate questions dataset is an open domain sentence pair dataset. It has more than 400,000 tagged sentence pairs formatted like "text1 text2 is duplicate" means whether the two sentences are semantically similar. If they are semantically equal, the tag will be "1", otherwise "0".
For the evaluation we choose Medical Question/Answer datasets ehealthforumQAs,questionDoctorQAs, and webmdQAs

For Selective dataset
train.csv

For full quora dataset
Quora Question Pairs(kaggle )

For evaluation datasets
Medical Question/Answer dataset

| id | qid1 | qid2 | question1 | question2 | is_duplicate |
|---|---|---|---|---|---|
| 447 | 895 | 896 | What are natural numbers? | What is a least natural number? | 0 |
| 1518 | 3037 | 3038 | Which pizzas are the most popularly ordered pizzas on Domino's menu? | How many calories does a Dominos pizza have? | 0 |
| 3272 | 6542 | 6543 | How do you start a bakery? | How can one start a bakery business? | 1 |
| 3362 | 6722 | 6723 | Should I learn python or Java first? | If I had to choose between learning Java and Python, what should I choose to learn first? | 1 |

### 5.1.1 Environment

We have experimented the deep learning models on Google Colab (Tesla P100-PCIE,16GB) to validate the semantic similarity between two sentences.

## 5.2 Experiments report

partial experiments report/total experiment count: approx. 100/Best experiment result with snapshot

Full Dataset = 240000

| Batch size | epochs | max seq | embeded dim | Model Drop ou | Hidden Layers | Reults |
|---|---|---|---|---|---|---|
| 1024 | 9 | 62 | 300 | 0.1 | 50 | 77% |
| 1024 | 6 | 62 | 300 | 0.1 | 50 | 78% |
| 512 | 25 | 62 | 300 | 0.1 | 50 | 77% |
| 512 | 20 | 62 | 300 | 0.1 | 50 | 76% |
| 64 | 25 | 62 | 300 | 0.1 | 50 | 77% |
| 64 | 20 | 62 | 300 | 0.1 | 50 | 78% |
| 64 | 15 | 62 | 300 | 0.1 | 50 | 78.50% |
| 64 | 12 | 62 | 300 | 0.3 | 100 | 79% |

Filtered balanced Dataset =10000

| Batch size | epochs | max seq | embeded dim | Model Drop ou | Hidden Layers | Reults |
|---|---|---|---|---|---|---|
| 1024 | 9 | 32 | 300 | 0.3 | 100 | 80% |
| 1024 | 15 | 32 | 300 | 0.3 | 100 | 81% |
| 512 | 10 | 32 | 300 | 0.3 | 100 | 81% |
| 512 | 15 | 32 | 300 | 0.3 | 100 | 80% |
| 64 | 25 | 32 | 300 | 0.3 | 100 | 80% |
| 64 | 15 | 32 | 300 | 0.3 | 100 | 82% |
| 64 | 10 | 32 | 300 | 0.3 | 100 | 82.50% |
| 64 | 6 | 32 | 300 | 0.3 | 100 | 83.00% |
| 64 | 6 | 32 | 300 | 0.5 | 100 | 82.55% |
| 64 | 6 | 32 | 300 | 0.2 | 100 | 83.4%-83.65% |
| 64 | 6 | 32 | 300 | 0.1 | 100 | 81.50% |
| 64 | 6 | 10 | 300 | 0.2 | 100 | 83% |

## 5.3 best experiment result

This is our best experiment result we got in the below snapshot Comparing to the research paper their Final result : max seq length 10,batch size 1024,n epoch 9,n hidden 50,Dropout 0.1 Average accuracy by 30 times experiments: 0.8043
Our best result showing enhancing in the performance :
- Removing stop enhance it to // - Update Max sequence enhance it from to // - Update dropout

```
Total params: 8,163,315
Trainable params: 1,130,115
Non-trainable params: 7,033,200
_________________________________________________________________
Epoch 1/6
125/125 [==============================] - 22s 58ms/step - loss: 0.1870 - accuracy: 0.7004 - val_loss: 0.1611 - val_accuracy: 0.762
Epoch 2/6
125/125 [==============================] - 4s 30ms/step - loss: 0.1274 - accuracy: 0.8210 - val_loss: 0.1476 - val_accuracy: 0.7840
Epoch 3/6
125/125 [==============================] - 4s 29ms/step - loss: 0.0789 - accuracy: 0.8967 - val_loss: 0.1491 - val_accuracy: 0.8015
Epoch 4/6
125/125 [==============================] - 4s 30ms/step - loss: 0.0514 - accuracy: 0.9352 - val_loss: 0.1517 - val_accuracy: 0.8085
Epoch 5/6
125/125 [==============================] - 4s 29ms/step - loss: 0.0341 - accuracy: 0.9589 - val_loss: 0.1484 - val_accuracy: 0.8105
Epoch 6/6
125/125 [==============================] - 4s 30ms/step - loss: 0.0270 - accuracy: 0.9684 - val_loss: 0.1389 - val_accuracy: 0.8365
Training time finished.
6 epochs in        40.72
```