

---

# Electronic Health Record Predictor

---

**Shailesh Kumar Jha**

Department of Computer Science  
Georgia State University  
Atlanta, GA 30303  
[sjha3@student.gsu.edu](mailto:sjha3@student.gsu.edu)

**Dheeraj Reddy Jeeru**

Department of Computer Science  
Georgia State University  
Atlanta, GA 30303  
[djeeru1@student.gsu.edu](mailto:djeeru1@student.gsu.edu)

## 1. Abstract

With the advancements in technology, one of the major breakthroughs has been in the health sector. We have been able to slowly win over major catastrophic diseases and increase the life expectancy of people in general. Often, keeping medical records of the patients and maintaining historical data helps the doctors in creating the medical history of the patient which in turn helps them to diagnose the future problems with the patient faster and with increased accuracy. It also makes it easier for the patient and the inspecting doctor in case he has been referred to another doctor. The new doctor can easily understand his medical history and tailor the diagnosis as per the patient's health needs. We propose a new way of predictive analytics using LSTM and seq2seq modelling which can help the doctors take notes in a faster way to help them save their time and attend more patients simultaneously with the help of historical patient data and deep learning modelling.

## 2. Introduction

Keeping this in mind about the importance of keeping a patient's historical data, the Electronic Health Record (EHR) as shown in pic 1 was created. It has an extensive record of the patient in discussion and his entire medical history which can be exchanged between different hospitals based on the patient's approval when he undergoes medical check-ups at different places. Realizing the importance and sensitivity of individual medical records, extensive care must be taken to maintain perfect secrecy of these records and they should only be shared with the doctors examining the patient with the patient's permission or be disclosed to the patient.

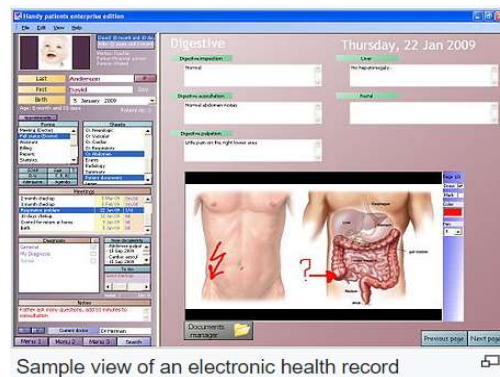


Fig 1

In tandem with these policies, it is essential for the medical practitioner to record every detail of the encounter between the patient and the advice and prescription offered by the doctor. Every visit must be recorded as per rule and each detail of the symptoms and diagnosis must be recorded in detail.

This becomes a cumbersome task for the doctors as they must juggle between noting down the important parts of their conversation while also working on the diagnosis based on the symptoms. Another important point of discussion is that when the discussion goes in depth, they must take care to not miss anything. The solution for this which many doctors practice, is to write everything on the go and be verbose in their details – a major reason for this being since they have people's lives at stake, and they do not want to risk it by virtue of missing important parts of their conversation.

Often, these discussions are repetitive and most of the information can be automated. Our goal is to help the doctors with this task of helping with the sentence prediction so that they can focus on their patients. If we can help them with the text prediction and with the probable medication for it, we can help the doctors help patient and in turn also help them spend lesser time with their patients. This can also increase the number of patients that they help every day.

We propose to create a smart Electronic Health Record prediction system which can help doctors with predictive and repetitive sentences and suggest them complete lines which they can either choose to accept or correct on the go. This will considerably reduce their time spent on taking EHR notes and work with the patients. We also propose to have a dynamic display of probability histogram which takes the symptoms and matches with the historical records to come up with predictions of various medications for it and their probability based on the data entered by the doctor. This prediction improves as we input more diagnosis and dynamically refreshes with time.

This will drastically improve over time as we have a bigger dataset to train our model and will be more efficient by applying various deep learning concepts. In summary, we want to achieve the following goals:

- a) Use deep learning methodology to help with predictive end of sentences.
- b) Feed the completed sentences to another model and predict conditional probability for various diseases based on it.
- c) Help doctors help more patients as they will need less time to fill EHR and work more with the patients.

### **3. Methodology**

At the helm of this concept is the usage of NLP (Natural Language Processing) used to process the historical dataset. The EHR dataset is very difficult to find since it contains confidential medical data. Creating the bag of words from this is the first part of our algorithm. We use the LMs (Learning Models) to learn a representation of a text corpus like word embedding but a better one. In simple word, it breaks up the text corpus and assigns probabilities to text sequences, typically one word at a time. To accomplish this, it uses the chain rule of probability. So, it first checks the probability of the first word and continues with the probability of the second

word given the first word and so on. Mathematically speaking, the probability of the sentence - “The dog barked at the stranger” will be given as shown below:

$$P(\text{"The dog barked at the stranger"}) = P(\text{"The"}) P(\text{"dog"} | \text{"The"}) P(\text{"barked"} | \text{"The", "dog"}) ..$$

$$P(w_{1:n}) = P(w_1) P(w_2 | w_1) P(w_3 | w_{1:2}) ..... P(w_n | w_{1:n-1})$$

Where n is the number of words.

A special case of generating an arbitrary length sequence given an arbitrary length of sequence as input is called conditioned generation as the output is conditioned on the input. This type of modelling is known as Seq2Seq modelling.

### 3.1 LM and RNN

RNN has emerged as the go to architecture for sequence modelling as it helps us overcome the shortcomings of Markov models.

For the prediction of sentences and words, we need to have a Learning model more granular than the ones at the word level, hence we propose to build a LSTM which is shown below (Fig 2). The Learning Model learns a representation of text like word embedding. In simple terms, we use it to input a word to generate a word or a sequence of word based on the historical learning dataset.

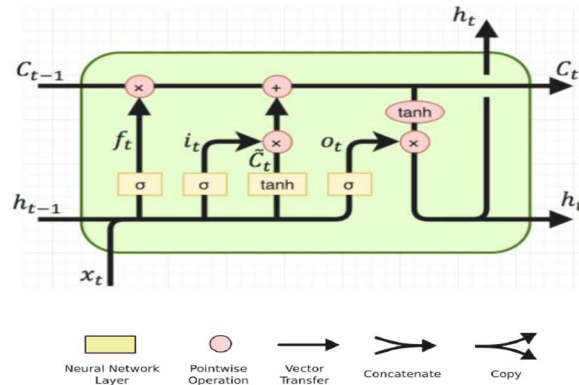


Fig 2

### 3.2 Seq2Seq

We use Seq2Seq first quoted by Google. The seq2seq model is shown in Fig 3 and consists of two components – encoder and decoder. Hence it is also sometimes known as encoder- decoder network. Here the encoder network uses deep neural networks and converts the input words to corresponding hidden vectors where each vector represents the current word and the context of the word. Similarly, the decoder takes the hidden vector as the input and then predicts the word.

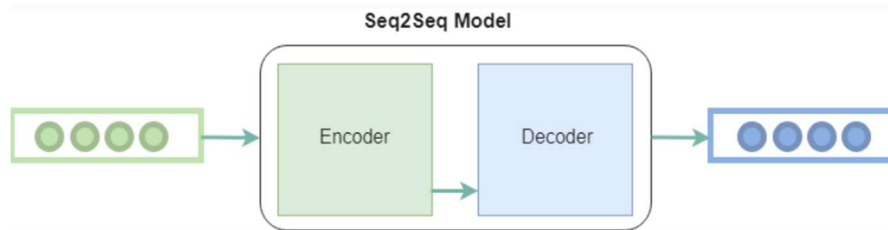


Fig 3

This model takes a sequence of words or sentences and generates another sequence of words or sentences using LSTM Recurrent Neural Networks (RNN). The logic behind it is the word association and the associated probabilities. It checks the probabilities of the sequence of word as you add words to the text and based on that it predicts the sequence of words or sentences which has the highest probability. This changes at real time with each addition of a new word and hence every addition might result or a similar or different suggestion.

If we also have access to previous health records of the patient, we can use that to further improve our suggestion using what is known as “conditional generation”. In this situation, it also considers the previous medical history to check symptoms or medicines which have worked for the patient before or dosage which might or might not have worked. For example, Diuretics might work on a high blood pressure patient and not have any side effects, but the same patient might suffer severe side-effects with Bet-blockers.

If we closely observe the Seq2Seq model with RNN and compare with the concept of conditioned generation, it is obvious that they both are relation with many to one many relations i.e. an input sentence of any length can generate an output sentence of any length. The many-to-many modelling is shown in fig 4

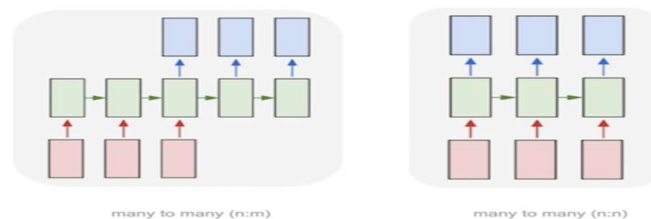


Fig 4

When you type a sentence, based on the existing sequence of words in the existing training dataset, they calculate the probability of the next few words or sentence and then the one with the highest probability is suggested. The author can choose to accept it or ignore and keep typing. At this point, with the addition of each extra word, this probability is again recalculated to project the new suggestion and this cycle keeps repeating until the end of the document.

#### 4. Related Work

We use the power of predictive analytics and the concept of Google’s Gmail smart compose functionality and used a combination of both of them on

medical and EHR data for smarter medical EHR text prediction.

## 5. Experiment

We are trying to help better predict medical sentences taking into consideration medical terminologies and jargons based on the symptoms of the patient. For this, we start with the regular input from the user where we ask about their condition. We have used the UCI dataset to get a list of symptoms and their reviews based on which we understand the issue with the patient and the treatment and the medication for him. This dataset is new as it was submitted in 2018 and has 215063 instances. Getting medical data is very difficult as it contains personally identifiable information.

We use the seq2seq modelling and the learning model to get the conditional probability of the words. We have used SoftMax activation function in our model and optimizer used has been Adam. While we were training the model, we realized that the training of the model is really expensive, and it can be confirmed from the fact that when we tried to run the model with a training dataset of 500 with 300 epochs which took over 20 hours. Here the accuracy achieved was 66.69%. We reduced our training model to a dataset of 400 with 300 epochs which took the 14 hours to complete and the results are discussed later in detail in this document. The generated efficiency in this case is 55.80%

Once the model has been trained and sorted based on their conditional probability, we also input the number of words that we want the model to predict and based on that, the output is generated.

For the design, we have used python and the associated libraries like Keras and tensorflow for training our model and employed it for the probability output. Then, based on the input and the number of words after it, the most probable output is generated.

### 5.1 Issues

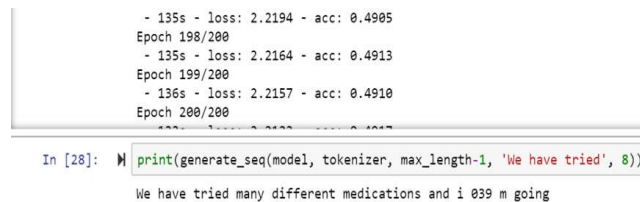
- 1) Since we are dealing with a patient's sensitive information, getting a dataset was difficult with the required input on the reviews and its probable side effects. We have found a dataset on UCI, but we are trying to improve our training model and a better dataset will help us output a more efficient result.
- 2) With medical jargons, it was difficult to map it to regular sentence association rules and try to use probabilities on it.
- 3) The python code does give us the output based on the conditional probability distribution, but it does not have a suitable UI element to show the prediction to the end user i.e. it is not end user ready. For this, we must create an interactive HTML page where the user inputs which is then taken as an input for the python code and the associated prediction is generated which is sent back to the HTML page in real time as output.
- 4) At times, the seq2seq modelling was difficult as the reviews did not always have proper sentence creation. Instead, they just had keywords, and, in that case, the association was difficult.

- 5) The training rate of the model is slow since the domain of sentence creation is huge, and the user can write anything. So, prediction becomes difficult.
- 6) As we worked closer towards our goal, we realized that the processing time for the dataset with our limited system capability is a bottleneck and we would require high computation power for faster processing.
- 7) The larger the dataset to train our model, the better will be the accuracy since this is different from regular sentence prediction as it contains medical terms also.

## 5.2 Experimental Result

As discussed above, we used various combination of training dataset and epochs to get various efficiencies. Based on this we are discussing few of our results.

- a) For 500 dataset and epoch 200. When we start the sentence with 'We have tried' and ask the model to predict the next 8 words, we get the following result



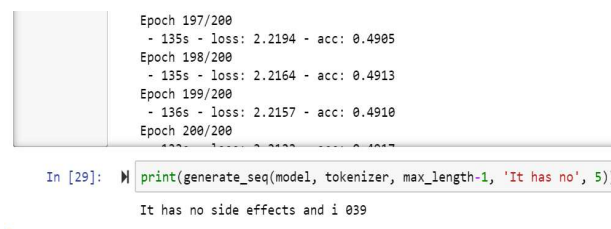
```

- 135s - loss: 2.2194 - acc: 0.4905
Epoch 198/200
- 135s - loss: 2.2164 - acc: 0.4913
Epoch 199/200
- 136s - loss: 2.2157 - acc: 0.4910
Epoch 200/200
In [28]: print(generate_seq(model, tokenizer, max_length-1, 'We have tried', 8))
We have tried many different medications and i 039 m going

```

Fig 5

- b) For 500 dataset and epoch 200. When we start the sentence with 'It has no' and ask the model to predict the next 5 words, we get the following result.



```

Epoch 197/200
- 135s - loss: 2.2194 - acc: 0.4905
Epoch 198/200
- 135s - loss: 2.2164 - acc: 0.4913
Epoch 199/200
- 136s - loss: 2.2157 - acc: 0.4910
Epoch 200/200
In [29]: print(generate_seq(model, tokenizer, max_length-1, 'It has no', 5))
It has no side effects and i 039

```

Fig 6

- c) For the next experiment, we used the model with 400 dataset and 200 epochs and gave 'I' as input with the task of predicting the next 5 words and the following result was recorded.



```

- 135s - loss: 2.2194 - acc: 0.4905
Epoch 199/200
- 136s - loss: 2.2157 - acc: 0.4910
Epoch 200/200
In [36]: print(generate_seq(model, tokenizer, max_length-1, 'I', 5))
I 039 ve been on this

```

Fig 7

- d) When we used the model with 400 dataset and 200 epochs and gave 'I am' as input with the task of predicting the next 5 words and the following result was recorded.

```
Epoch 198/200
- 135s - loss: 2.2164 - acc: 0.4913
Epoch 199/200
- 136s - loss: 2.2157 - acc: 0.4910
Epoch 200/200
- 135s - loss: 2.2164 - acc: 0.4913

In [38]: print(generate_seq(model, tokenizer, max_length-1, 'I am ', 5))
I am now on it for a
```

Fig 8

- e) When we used the model with 400 dataset and 200 epochs and gave 'I was' as input with the task of predicting the next 5 words and the following result was recorded.

```
Epoch 198/200
- 135s - loss: 2.2164 - acc: 0.4913
Epoch 199/200
- 136s - loss: 2.2157 - acc: 0.4910
Epoch 200/200
- 135s - loss: 2.2164 - acc: 0.4913

In [39]: print(generate_seq(model, tokenizer, max_length-1, 'I was ', 5))
I was prescribed fetzima transitioned off cymbalta
```

Fig 9

## 6. Conclusion

Based on the above results, we can say that our model is working but we can achieve greater accuracy when we have the computational power to run it for more dataset and for higher epochs.

## 7. Future Scope

- We can use a HTML page with REST API to show this on a browser with a browser embedded editor to show how the prediction model works.
- We then use LMs and char n gram along with Seq2Seq model to determine the prediction of next group of words even with half written words.
- We plan to also feed the completed sentence back into our model to display dynamic probability histogram of various medicines that can fit the description of the input data.
- Use demographic data like current weather conditions and topological information to better tailor the data and give more accurate predictions.

## 8. References

- [1] <https://towardsdatascience.com/gmail-style-smart-compose-using-char-n-gram-language-models-a73c09550447>.
- [2] Fig 1([https://en.wikipedia.org/wiki/Electronic\\_health\\_record](https://en.wikipedia.org/wiki/Electronic_health_record)).

[3] Fig 2 (<https://towardsdatascience.com/gmail-style-smart-compose-using-char-n-gram-language-models-a73c09550447>).

[4] Dataset : <https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+%28Drugs.com%29>

[5] <https://towardsdatascience.com/introduction-to-language-models-n-gram-e323081503d9>

[6] <https://www.geeksforgeeks.org/seq2seq-model-in-machine-learning/>

[7] <https://ehrintelligence.com/news/google-study-uses-entire-patient-ehr-for-predictive-analytics>