International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 5, Issue 8, ISSN No. 2455-2143, Pages 263-268
Published Online December 2020 in IJEAST (http://www.ijeast.com)

ENTITY EXTRACTION FROM UNSTRUCTURED MEDICAL TEXT

G Deepank
Department of Computer Science and Engineering
PES University, India

R Tharun Raj
Department of Computer Science and Engineering
PES University, India

Aditya Verma
Department of Computer Science and Engineering
PES University, India

***Abstract*—Electronic medical records represent rich data repositories loaded with valuable patient information. As artificial intelligence and machine learning in the field of medicine is becoming more popular by the day, ways to integrate it are always changing. One such way is processing the clinical notes and records, which are maintained by doctors and other medical professionals.**

Natural language processing can record this data and read more deeply into it than any human. Deep learning techniques such as entity extraction which involves identifying and returning of key data elements from an electronic medical record, and other techniques involving models such as BERT for question answering, when applied to all these medical records can create bespoke and efficient treatment plans for the patients, which can help in a swift and carefree recovery.

Keywords—Entity, Features, Extraction, NLP, BERT

I. INTRODUCTION

There have been many documented occurrences of medical practitioners making the wrong call on a specific disease or ailment of a patient, that has led to the wrong diagnosis, and thus the wrong treatment. Thus all the stakeholders in the process, suffer a loss, and most importantly, the patient may not eventually be cured and will have to suffer. To combat this issue, we propose this solution which can aid doctors, nurses, and other medical staff to make the right diagnosis as fast as possible, with high accuracy. We aim to use NLP and Deep Learning for extracting disease labels and other ground truth data from complicated unstructured clinical reports and doctor notes (which is often overlooked) and converting it to a structured format for efficient diagnosis. There are some unavoidable challenges, apart from programming, such as how different doctors have different ways of writing reports. Thus, using NLP and other associated technologies, we can interpret these notes and conclude the diagnosis. In this project, we will make use of a rules-based technique for this task. The effects of negative and uncertain mentions on the same will be observed, assessed and evaluated using the F1 score metric from the large dataset of the Electronic Medical Reports. Finally, the BERT language model is used to build a simple question answering system, an extension of entity extraction, which can help to get a better understanding of the ailment. Using the above-mentioned techniques, we hope to assist medical professionals to make better diagnoses, thus benefiting the patient.

II. LITERATURE SURVEY

A detailed literature survey was done on this project. We found three such papers, and their findings have been listed below:

A. *[1]Leveraging word embeddings and medical entity extraction for biomedical dataset retrieval using unstructured texts:*

- Proposed an IR system for biomedical dataset retrieval.
- Does not take into consideration negative and uncertain mentions.
- The workflow of the entire project was very unstructured and complex, thus it was not desirable.

B. *[2]A Supervised Named-Entity Extraction System for Medical Text:*

- Developed a supervised linear-chain Conditional Random Fields (CRF) model using 10-fold cross-validation on the training dataset.
- The EMR documents used were present in a well-structured form with a header, document body, and a footer. Since the header and the footer contain information that is useful only for clinical administration and not diagnosis, the analysis was performed only on the document body.
- Failed to identify the non-contiguous named entities.
- Produced a bad F1 score `and recall.

C. *[3]CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning:*

- Developed a 121-layer convolutional neural network from scratch trained on the open source ChestX-ray14 dataset.
- Detects pneumonia from the given dataset at a higher precision than practicing radiologists.
- It takes a chest X-ray image as input and outputs the probability of pneumonia along with a heatmap localizing the areas of the image most indicative of pneumonia.
- The model and the radiologists were not permitted to use the patient's history (clinical notes, etc) which

has been shown to decrease radiologist diagnostic performance in interpreting chest radiographs.

Studying these sets of papers, we found that there are many areas to take inspiration from, as well as many areas where the implementation can be improved, such as:

- Improving model performance
- Reducing the overall complexity of the project
- Take negative and uncertain mentions into consideration.

III. Model Design

We have based our basic model design on the papers listed above, as well as some new things. Following is the approach that we have used to do this project:

A. *Exploratory Data Analysis*

The dataset we worked with essentially contained medical reports of numerous patients with an overall report impression. In addition to this, it contained 11 possible medical conditions that indicated whether a patient had tested positive or negative for each condition. In trying to understand the dataset, the following graphs and charts were plotted-

1. Distribution of abnormalities

A bar plot was made to observe the distribution of the various medical abnormalities.

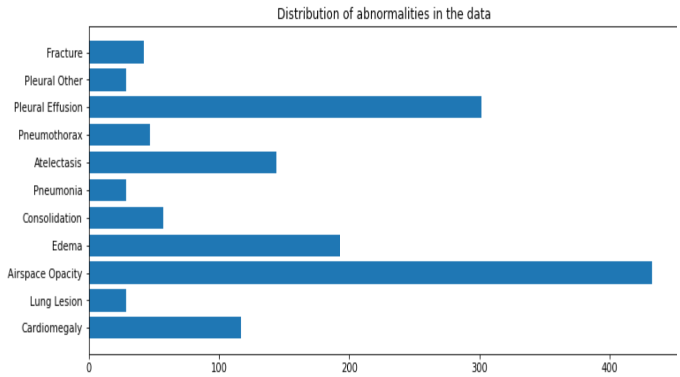

Fig. 1. Barplot showing distribution of abnormalities

It can be seen from the graph that the majority of the patients seem to have the following 5 medical abnormalities, Airspace Opacity, Pleural Effusion, Edema, Atelectasis, and Cardiomegaly.

2. Word-Cloud

A word-cloud plot was designed to see all the important words that occurred in the medical reports:

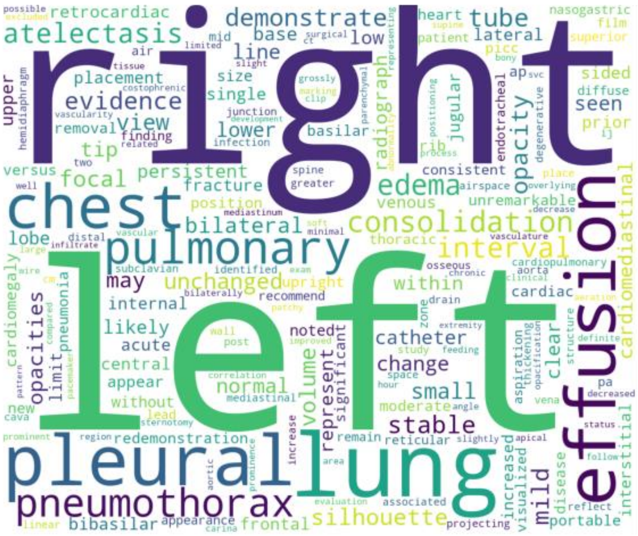

Fig. 2. Word-Cloud

The word-cloud suggests that the major words appearing were **'right'** and **'left'**, possibly indicative of the right and left lung. A large number of medical conditions along with doctors' remarks such as **'mild'** and **'stable'** are also profound.

3. Sentiment Polarity

Sentiment Polarity plots are made when the overall nature of the statement(s) has to be found. It can either be an overall positive statement, e.g: Good bone structure, or an overall negative statement, e.g: Bad bone structure. The sentiment polarity tool 'TextBlob' returns a float value, indicating to which side the statement is leaning.

A distribution plot on Sentiment Polarity of the report impressions was made:

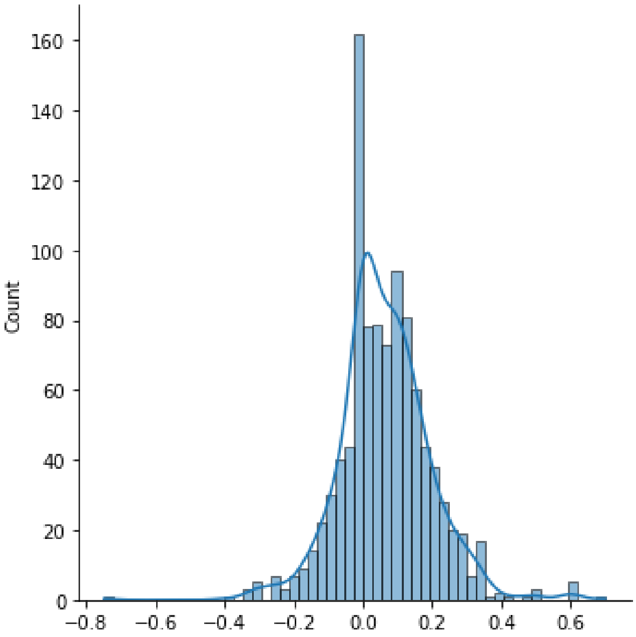

Fig. 3. Distplot of Sentiment Polarity

It can be seen that the remarks are generally neutral to positive. It is suggestive that the reports were possibly written this way, keeping the patients' mental well-being in mind.

B. Entity Extraction

A rules-based method was used for extracting the entities from the dataset. A rules-based method is where a set of predefined labels is added to a list.

- Included with the main dataset, is a set of synonyms for diseases and ailments, which various other doctors and medical professionals may use while noting the symptoms.
- The dataset is iteratively checked and finds the word in the synonyms list for every label by checking every word in the report impression. A report impression is simply an instance of the dataset.
- A dictionary is initialized, with the ailment as key and a flag as value, set to boolean false.
- If a synonym is found for the particular ailment, or the ailment itself is found in the report impression, then it sets its corresponding flag to true.
- A dictionary is returned that maps each category to a boolean value, which indicates the presence or absence of the abnormality.

C. Finding Negative and Uncertain Mentions

In some of the papers studied while doing the initial research for this project, we found that some of them didn't consider negative or uncertain mentions in the report impression. A simple example can be shown as:

Report Impression: No Pleural Effusion.

Model output: {'Pleural Effusion':true}

This error occurred due to not taking into consideration the presence of the word 'No' in the report, clearly telling that the patient does not have Pleural Effusion.

Thus, to take the negative and uncertain mentions into consideration, we follow the following steps:

- Defined a list containing predefined negative mentions, such as 'No', 'not'," didn't", etc.
- Iteratively traverse the report impression, matching each word to the ailment, or the negative mention.
- If a negative mention is found in the impression, then a flag is set to true, indicating the presence of a negative mention.

Now the output would be as follows:

Report Impression: No Pleural Effusion.

Model output: {' Pleural Effusion':False}

This will clear up the false positive cases in most of the report impressions.

D. Evaluating the performance of the model

In the analysis of classification, the F1-score is a measure of a test's accuracy. It is calculated by using the precision and recall of a test, where the precision is the number of correctly identified positive outcomes divided by the number of all positive outcomes (including false positives),and the recall is the number of correctly identified positive outcomes divided by the number of all samples that should have been identified as positive outcomes.

The F1 score is calculated by taking the geometric mean of the two values.

During the testing of our model, the F1 scores were calculated from a user-defined function by taking into consideration all relevant parameters. Surprisingly, the F1 score of some of the entities *decreased* after considering the negative mentions in the report. This was not expected as the diagnosis was improved in our test cases.

Following is a table showing the differences in F1 scores in one of our test cases:

***Report Impression*:**
normal heart size and pulmonary vascularity. No focal consolidation, pleural effusion, or pneumothorax. bones are unremarkable.

***Retrieved labels before considering negative mentions*:**
Airspace Opacity: False
Atelectasis: False
Cardiomegaly: True
Consolidation: True
Edema : False
Fracture: False
Lung Lesion: False
Pleural Effusion: True
Pleural Other: False
Pneumonia: False
Pneumothorax: True

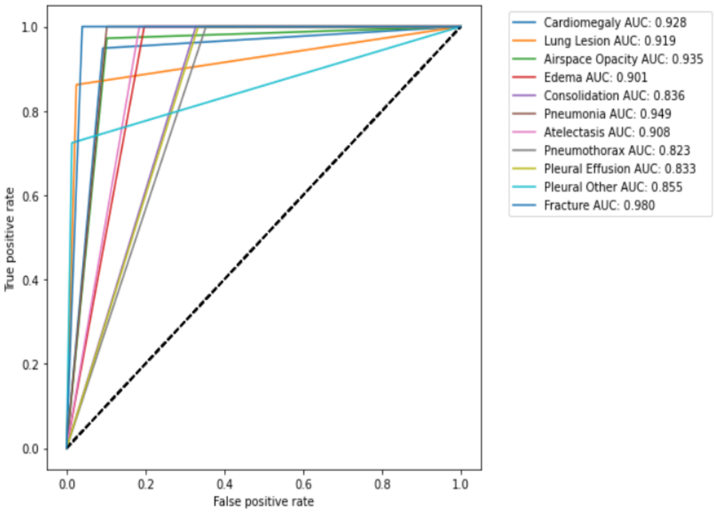

Fig. 4. ROC-AUC curve before considering negative mentions

Also, ROC-AUC score is calculated and plotted to visualize the model performance. It can be seen the individual model performance overall is fairly good. However, some labels have been identified incorrectly. As a result the diagnosis is incorrect. Now,

***Retrieved labels after considering negative mentions*:**
Airspace Opacity: False
Atelectasis: False
Cardiomegaly: False
Consolidation: False
Edema : False
Fracture: False
Lung Lesion: False
Pleural Effusion: False
Pleural Other: False
Pneumonia: False
Pneumothorax: False

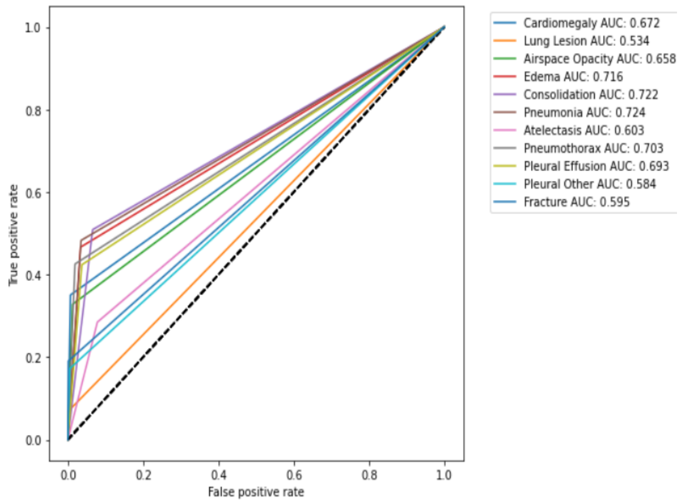

Fig. 5. ROC-AUC curve before considering negative mentions

When considering negative mentions in the report, the model performance surprisingly went drastically *down*. This can be seen in the corresponding ROC-AUC score and the F1 score. It was also observed that some of the labels among the entities performed slightly better in this case.

The corresponding F1 scores for this impression can be seen in the table below:

F1 SCORES

Entity	***Before Negative Mentions***	***After Negative Mentions***	***Difference***
Cardiomegaly	0.718	0.500	0.218
Lung Lesion	0.641	0.125	0.516
Airspace Opacity	0.923	0.488	0.435
Edema	0.708	0.581	0.127
Consolidation	0.270	0.392	-0.122
Pneumonia	0.369	0.364	0.005
Atelectasis	0.646	0.325	0.321
Pneumothorax	0.218	0.471	-0.253
Pleural Effusion	0.722	0.561	0.161
Pleural Other	0.667	0.256	0.411
Fracture	0.689	0.314	0.375
Average	0.597	0.398	0.199

This difference in F1 scores could be due to the following reasons:

- The number of negative mentions added to the initial negative mentions list was not enough.
- Negative mentions not being considered if it is present *after* the ailment. e.g.: “Pleural Effusion not present” may return a true value.

IV. QUESTION-ANSWERING MODEL USING BERT

Question answering (QA) systems is an area of NLP that has applications in tasks such as entity extraction and information retrieval. In this project, we will use BERT to build a simple QA system. BERT is a transformer-based model developed by researchers at Google. It achieves state-of-the-art accuracies in many NLP tasks.

In this project, for the sake of brevity, we have used a pre-trained BERT model (trained on the SQuAD dataset from Stanford University) and the concept of transfer learning to build a simple and effective QA system.

The method implemented to build a QA system for our medical application here is as follows:

A. *Installing ‘transformers’ library*

As mentioned above, we are going to use an advanced p-re-trained BERT model provided by *huggingface*. The **transformer** library has a large collection of pre-trained models for specific NLP tasks. It is simple to use and comes with TensorFlow and PyTorch as backends.

B. *Tokenizing*

Tokenizing is the process of assigning word embeddings to all the words present in the text. This varies from model-to-model. In this application, we are using the *‘Fine tuned BERT-large*’ question answering model. We load both the tokenizer and model weights to a variable using this pre-trained model.

C. *Pre-processing*

The input text to the QA model contains two components, namely the passage component, which is the medical report/clinical notes of the patient, and the questions component, which is a list of the questions asked. The list of questions can be changed according to the convenience of the stakeholders.

The two components are concatenated and fed into the model as a single vector, using the **[CLS]** token to specify the start of the questions, the two **[SEP]** tokens to define the start and end of the passage.

When viewing the tokens, it can be seen that the tokenizer takes every word and assigns tokens to them from the model, and also breaks down continuous words into two separate words. e.g: ‘classifying’ is broken down into ‘classify’ and

‘###ing’ for better model performance. ‘###’ at the start of any word means that the previous word is a part of it.

D. Modelling

The following steps are followed for the modeling phase:

1) *Segmentation*
BERT has two special segment embeddings which it uses to distinguish the passage and question tokens and finds the correct answer by adding the special embeddings to the both passage and question tokens. This process is handled by the transformers library. All we have to do is find the index of the first [SEP] token to get the length of the segments and create a mask of 0’s and 1’s for each token.
2) *Evaluation*
A tensor of the input token IDs and the segmentation IDs is fed to the pre-trained BERT model. It returns the start and end scores of the answer to be constructed.
3) *Answer construction*
The index of the highest start and end scores returned by the model are recorded. We pick the word as the answer which has the highest total score (provided end >= start). If the word is part of a continuous word, e.g: ‘###ing’, then it is combined with the previous word to combine to the original word. This is then added to the answer.

4) *Interface*
The passage is passed along with each question into the model and the answers are printed in an iterative manner, giving the viewer, who is a stakeholder such as a medical professional like a doctor, surgeons, family members etc. the condition of the patient at hand.

An instance of the model working is demonstrated below on an illustrative clinical abstract.

Passage provided:
"The patient is a 54 year old female. She complains of extensive skin roughness and sore legs. The patient also has bleeding in her gums. Patient says she is experiencing such symptoms for the first time. This consultation is a follow up based on the results of the report and for further evaluation. There is a deficiency of Vitamin C in her report. Possibly has scurvy. Heart rate is high. Blood sugar level is low. "

Questions asked:
"How old is the patient?"
"Does the patient have any complaints?"
"What is the reason for this consultation?"
"What does her report show?"
"What other symptoms does the patient have?"

Output obtained:
Question 1: How old is the patient?
Answer: "54"

Question 2: Does the patient have any complaints?
Answer: "the patient complains of skin roughness and sore legs"

Question 3: What is the reason for this consultation?
Answer: "further evaluation"

Question 4: What does her report show?
Answer: "deficiency in Vitamin C"

Question 5: What other symptoms does the patient have?
Answer: "bleeding gums"

As we can see, by using the BERT model, we could construct a fairly accurate and ideal QA model which can answer the provided questions in simple language. For obtaining better performance, a sequence model trained explicitly on clinical data and medical records may help.

V. ACKNOWLEDGEMENT

We would like to express our sincerest gratitude to Prof. Jyothi R ma’am for her continued support, guidance and encouragement. We also thank PES University for letting us take up the course ‘Special Topic - Mini-Project’ and giving us the opportunity to work on this project.

VI. REFERENCES

[1] Wang, Yanshan, Rastegar-Mojarad, Majid, Komandur-Elayavilli, Ravikumar and Liu, Hongfang. Leveraging word embeddings and medical entity extraction for biomedical dataset retrieval using unstructured texts.

[2] Bodnari, Andreea, Deleger, Louise, Lavergne, Thomas Neveol, Aurelie, and Zweigenbaum, Pierre. A Supervised Named-Entity Extraction System for Medical Text.

[3] Rajpurkar, Pranav, Irvin, Jeremy, Zhu, Kaylie, Yang, Brandon, Mehta, Hershel, Duan, Tony, Ding, Daisy, Bagul, Aarti, Ball, Robyn L., Langlotz, Curtis, Shpanskaya, Katie, Lungren, Matthew P. and Ng, Andrew Y. CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning.

[4] Rajpurkar, Pranav, Hannun, Awni Y, Haghpanahi, Masoumeh, Bourn, Codie, and Ng, Andrew Y. Cardiologist-level arrhythmia detection with convolutional neural networks.

[5] Islam, Mohammad Tariqul, Aowal, Md Abdul, Minhaz, Ahmed Tahseen, and Ashraf, Khalid. Abnormality detection and localization in chest x-rays using deep convolutional neural networks.

[6] Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai, and Fei-Fei, Li. Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern

Recognition, 2009. CVPR 2009. IEEE Conference on, pp. 238–265. IEEE, 2009.

[7] Grewal, Monika, Srivastava, Muktabh Mayank, Kumar, Pulkit, and Varadarajan, Srikrishna. Radnet: Radiologist level accuracy using deep learning for hemorrhage detection in ct scans.

[8] Huang, Gao, Liu, Zhuang, Weinberger, Kilian Q, and van der Maaten, Laurens. Densely connected convolutional networks.

[9] NegBio library:

Available: https://github.com/ncbi-nlp/NegBio

[10] Pre-trained BERT model used for building question answering system

Avalable:https://huggingface.co/bert-large-uncased-whole-word-masking-finetuned-squad

[11] Medical Transcription Sample Report (mtsamples.com)

Passage-1,Available:https://www.mtsamples.com/site/pages/sample.asp?Type=6-Cardiovascular%20/%20Pulmonary&Sample=1597-Abnormal%20Echocardiogram

[12] Medical Transcription Sample Report (mtsamples.com) Passage-2,Avalable:https://www.mtsamples.com/site/pages/sample.asp?Type=18-Dermatology&Sample=407-Wasp%20Sting%20-%20SOAP