Disease Diagnosis

Natural Language Processing and Classification Models

[Project Report] – By Aakriti Pandey, Ruben Maharjan, and Shirish Khanal

Introduction

Disease diagnosis is a classification project that classifies diseases based on symptoms provided. It is a twofold project where we aim to extract medical keywords from a medical journals' and put it through a classification model to diagnose the disease. Medical illnesses are diagnosed by healthcare professionals based on symptoms. A tool that can do the same would aid the healthcare professional in the process of diagnosis. The end model could also be used by patients to get a diagnosis when doctors are not available through medical chatbots.

Problem statement

Health care is not accessible to all. If we could have an application that could take symptoms from patients and diagnose disease, it would be helpful. Storing a list of symptoms and having the system diagnose the disease is not enough. There could be several factors related to a disease. We need to find reliable literature sources and analyze plain texts to diagnose a disease accurately.

Motivation

One of the major motivations for this project is the poor health care system in Nepal and the unavailability of doctors in rural communities. Also, doctors do not have expertise in multiple

domains. We wanted to research NLP (Natural Language Processing) and explore classification models to predict diseases from text data.

Research questions the project will answer

This project aims to evaluate different machine learning models for text classification and find the best in terms of accuracy. We also aim to evaluate the possibility of using abstracts from medical journal to classify doctor-patients conversation into their possible diseases.

Literature review

Among various literature, 'Disease prediction from various symptoms using machine learning' (Keniya, et al., 2020) matched ours the most. It uses symptoms and other user information to classify diseases. The dataset used by Keniya, et al. contained more than 230 diseases and associated symptoms. Various machine learning approaches like Decision Tree, Naïve Bayes, and KNN (K-Nearest Neighbors) were implemented. KNN gave the best results with 93.5% accuracy. Another Disease Diagnosis System that we found is Medical Diagnostic System (Shortliffe), it uses google search and related articles to diagnose the disease based on symptoms. It searches for the exact same text given by a user.

Novelty of the work

Our project is unique since the dataset we are using is from authentic medical journals and it uses large dataset i.e., more than 40,000 records. Natural Language Processing has never been used to extract insights about symptoms.

Description of Dataset

Our primary source of training data is PubMed, a biomedical literature library. PubMed allows the user to explore the medical literature based on keywords. In our case, we fed disease name as a keyword, which resulted in thousands of medical journals related to that disease. We did this by querying the database using the *pymed* library. As we are primarily concerned with symptoms, we modified the query to return journals that also include the words, “symptoms”, “experiencing”, “presence of”, “sign off”, “suggestion”, “clue”, and “hint of”. We tested the query to include each disease’s name and each of these words individually to check for the accuracy of each. From this, we get a CSV file with columns: “Abstract” and “Disease”.

Example of raw data:

Abstract	Disease
Efforts to eradicate are largely threatened by drug-resistant , particularly, multidrug-resistant (MDR-TB). Screening and identification potential biomarkers for MDR-TB is crucial to diagnose early and reduce the incidence of MDR-TB. To screen the differentially expressed long non-coding RNAs in MDR-TB, the lncRNA and mRNA expression profiles in serum derived from healthy controls (HCs), individuals with MDR-TB and drug-sensitive (DS-TB) were analyzed by microarray assay and 10 lncRNAs were randomly selected for further validation by reverse transcription-quantitative real-time PCR(RT-qPCR). The biological functions of differentially expressed mRNAs as well as relationships between genes and signaling pathways were investigated using Gene Ontology (GO) and the Kyoto Encyclopedia of Genes and Genomes (KEGG), respectively. A total of 353 differentially expressed lncRNAs (312 upregulated) and 202 mRNAs (99 upregulated) were found in the MDR-TB group compared to HCs. And compared with the DS-TB group, 442 differentially expressed lncRNAs (115 upregulated) and 190 mRNAs (87 upregulated) were found in the MDR-TB group. The expression levels of lncRNA n335659 were found to differ significantly between each group by RT-qPCR. Compared with DS-TB group, the GO analysis showed that the differential mRNAs were mainly enriched in the processes associated with the detection of the chemical stimulus, the regulation of mRNA metabolic process and neutrophil activation in the MDR-TB group; the KEGG analysis indicated that the differential mRNAs between DS-TB and MDR-TB were mainly enriched in proteasome and Notch signaling pathway, which might reveal a fraction of the mechanism of MDR-TB. The discovery of the serum lncRNA n335659 might serve as a potential biomarker for MDR-TB and Notch signaling pathway provided a new clue for the investigation of the pathological mechanism of MDR-TB.	tuberculosis
is an infectious disease that involves any organ. However, the primary pituitary is an extremely rare disease. Intracranial tuberculomas account for 0.15-5% of intracranial space-occupyin	tuberculosis
While chest CT provides important clue for diagnosis of miliary (TB), patients are occasionally missed on initial CT, which might delay the diagnosis. This study was to evaluate the clinical ar	tuberculosis
This study investigated the potential role of paracardiac fat stranding (FS) interspersed with multiple fluid collections (FC) as a clue to differentiate between pleural (pleural TB) and maligna	tuberculosis
An article presented by Peterson and Verkhratsky in 2016 explore the linkage of calcium and ATP for a themed issue of "Evolution brings Ca	tuberculosis
is a leading cause of death worldwide. BCG is an effective vaccine, but not widely used in many parts of the world due to a variety of issues. Mycobacterium vaccae (M. vaccae) is another vaccine used in human subjects to prevent . In the current study, we investigated the potential mechanisms of M. vaccae vaccination by determining differentially expressed genes in mice infected with M. before and after M. vaccae vaccination. Three days after exposure to M. H37Rv strain (5 × 10 M. vaccae vaccination provided protection against M. infection (most prominent in the lungs). We identified 2326 upregulated and 2221 downregulated genes in vaccinated mice. These changes could be mapped to a total of 123 signaling pathways (68 upregulated and 55 downregulated). Further analysis pinpointed to the MyD88-dependent TLR signaling pathway and PI3K-Akt signaling pathway as most likely to be functional. M. vaccae vaccine provided good protection in mice against M. infection, via a highly complex set of molecular changes. Our findings may provide clue to guide development of more effective vaccine against .	tuberculosis

Approach

Data collection

As the number of diseases is huge, it is nearly impossible to create a predictive model for each. Hence, we focused on some largely prevailing diseases and focused on collecting data on them. The diseases are: 'diabetes', 'hypertension', 'arthritis', 'tuberculosis', 'pneumonia', 'peptic ulcer', and 'gastroenteritis'. Since our primary source of data is '*abstract'* from PubMed, we developed a python script to extract abstracts that also contains words similar to symptoms. This script generated a CSV file with two columns: abstract and disease.

Data cleaning

We intend to build a model using medical journals; hence, a large part of our work is Natural Language Processing (NLP). Owing to that, our data cleaning process also focused on NLP-specific techniques. The PubMed database returned a huge list of data with a lot of features. We were only interested in the Abstract. We cleaned this initial dataset we obtained from PubMed and created a CSV file that had two columns: disease, and abstract.

For the abstract column, we removed punctuation and stop words. Punctuation might be an important feature of language but does not add value to NLP models. Stop words are the most common words in a language, but they do not have relevance in context. We used a list of 179 stop words from the 'nltk' library and removed them. Further, we also used 'nltk' to tokenize and then lemmatize the abstract column. Tokenization is the process of converting long text into discrete forms, which makes it easier to calculate the occurrence of each word in the document. Lemmatization is the process of converting words to their root form. It is used when the meaning of words is not important. It is done to save resources.

Feature extraction:

Since both columns, *'abstract'* and *'disease'*, are crucial for our model there is no need to reduce dimensions. One of the redundant features in our dataset is the repetition of certain words which we do not need like 'Disease', 'Diagnosis' etc. Using such words in our prediction model is time-consuming and can cause bias as it will have the same weight as other important symptom-related words. So, to solve this problem we used TF-IDF (Term Frequency-Inverse Document Frequency).

Feature selection

We used TF-IDF (Term Frequency-Inverse Document Frequency) to solve the problem of having repetitive words that we do not need. It quantifies the importance of words in a corpus. It is the product of term frequency in a sentence and inverse document frequency i.e., the number of times the word appears in the whole document.

Model building

The models we used and evaluated in the projects are as follows:

a. Logistic Regression
b. (Multinomial) Naive Bayes
c. Linear Support Vector
d. Random Forest
e. Bidirectional Recurrent Neural Network (BRNN)

f. Convolutional Neural Network (CNN)

a) Logistic Regression: Logistic regression model exploits the relationship between dependent and independent variables in the dataset. In our case, it is the frequency of different symptoms (independent variable) and disease category (dependent variable) in the text.

b) (Multinomial) Naïve Bayes: Naïve Bayes model is based on the mathematical model of conditional probability. It calculates the probability that a data point belongs to a particular class. In our case, the probability is calculated based on the occurrence of different symptoms in the text.

c) Linear Support Vector Machine(SVM): Linear support vector machine draws hyperplanes on the graph to distinctly divide data points into different classes. The number of hyperplanes depends on the number of classifications, in our case, it is 7, the number of diseases.

d) Random Forest: Random Forest is made up of a large number of decision trees, where each decision tree gives a prediction and the class with the most votes becomes the decision of the model.

e) Bidirectional Recurrent Neural Network (BRNN) with Long Short-Term Memory (LSTM): RNN are neural networks that work great with time-series or sequential data. We have implemented a Bidirectional RNN (BRNN) with LSTM. The BRNN considers both the previous data and future data to get the context. Our RNN currently has two BRNNs with

LSTM. Besides the two BRNN's the input words go through an embedding layer that transform words into vectors which are fed into the BRNN's. We have used "softmax" as our activation function. This activation function allows us multiclass classification.

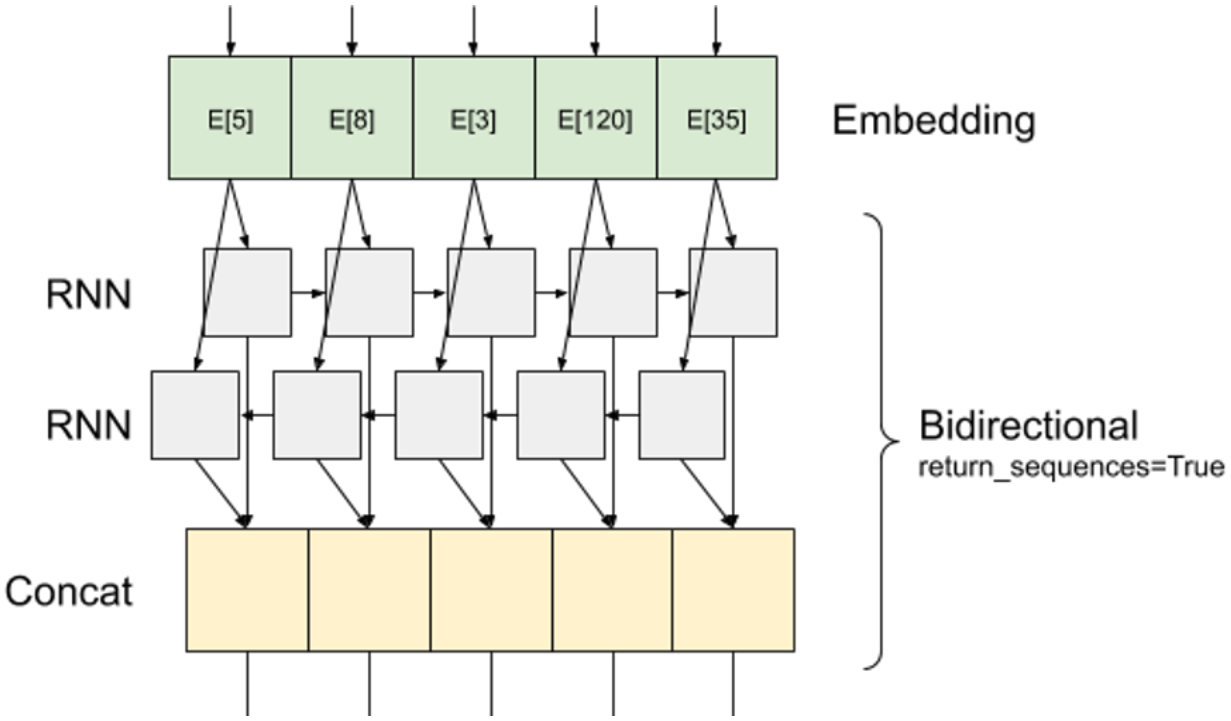

Figure 1: Architecture of BRNN

f) Convolutional Neural Network (CNN): We have implemented CNN with 1 Dimension input vector to classify our text. Convolutional neural network with two layers has been implemented. Predecessor to these layers is an Embedding layer. We have used the parameters that achieved the highest accuracy based on our manual training of the model with random parameters.

Results and insights

We implemented and evaluated 6 classifiers. We found that Linear Support Vector Machine is the best, accuracy wise. SVM generally performs better in text classification when there is interaction between datapoints. For example: symptoms for same disease in different abstract often repeat, i.e., similar symptoms occur together. SVM is good at capturing such interaction. Hence, we believe SVM outperformed other models. Classification report also shows high f1-score for SVM. f1 score combines precision and accuracy. Similarly, confusion matrix shows high true positive for most categories in SVM model.

Results for all models are shown in the ***graph*** below:

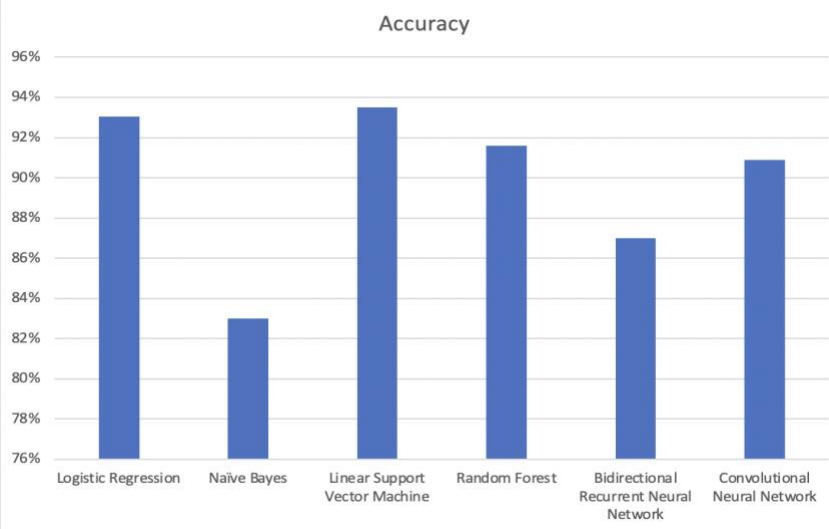

Figure 2: **Accuracy of different models.** The above chart shows the accuracy of all the models tested. SVM is the most accurate with accuracy score of 93.5%.

	precision	recall	f1-score	support
arthritis	0.93	0.90	0.91	784
diabetes	0.78	0.90	0.84	829
gastroenteritis	1.00	0.38	0.55	267
hypertension	0.87	0.88	0.87	794
peptic ulcer	1.00	0.11	0.20	168
pneumonia	0.76	0.90	0.82	805
tuberculosis	0.84	0.91	0.88	809
accuracy			0.84	4456
macro avg	0.88	0.71	0.73	4456
weighted avg	0.85	0.84	0.82	4456

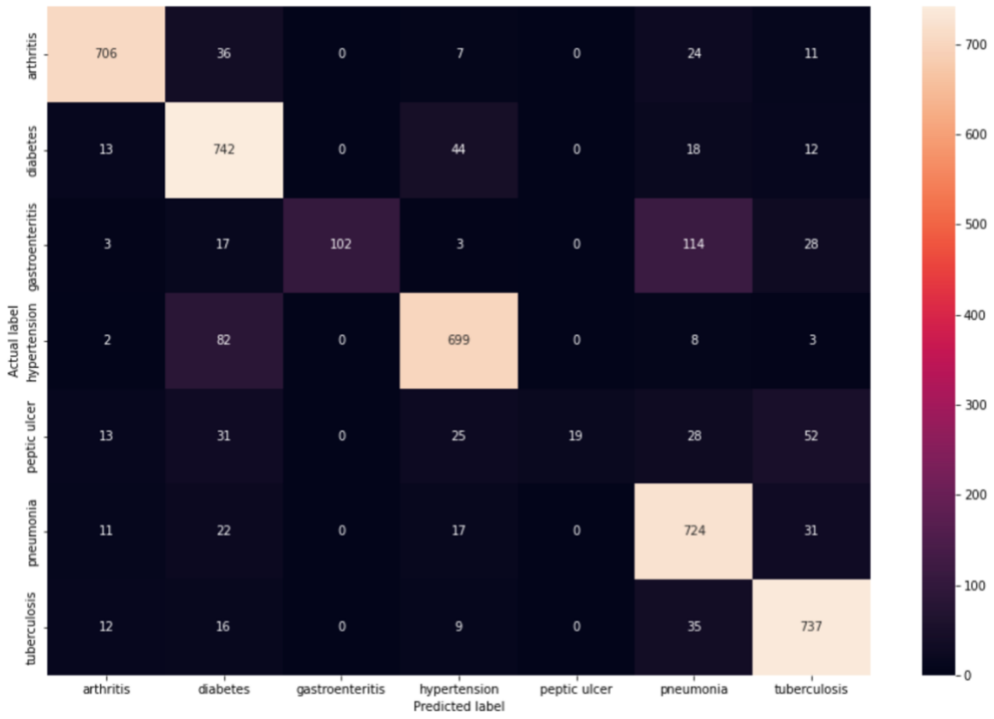

Figure 3: Classification Report and Confusion matrix for Naïve Bayes Model

	precision	recall	f1-score	support
arthritis	0.97	0.96	0.97	784
diabetes	0.93	0.93	0.93	829
gastroenteritis	0.99	0.93	0.96	267
hypertension	0.94	0.96	0.95	794
peptic ulcer	0.98	0.94	0.96	168
pneumonia	0.91	0.93	0.92	805
tuberculosis	0.94	0.94	0.94	809
accuracy			0.94	4456
macro avg	0.95	0.94	0.95	4456
weighted avg	0.94	0.94	0.94	4456

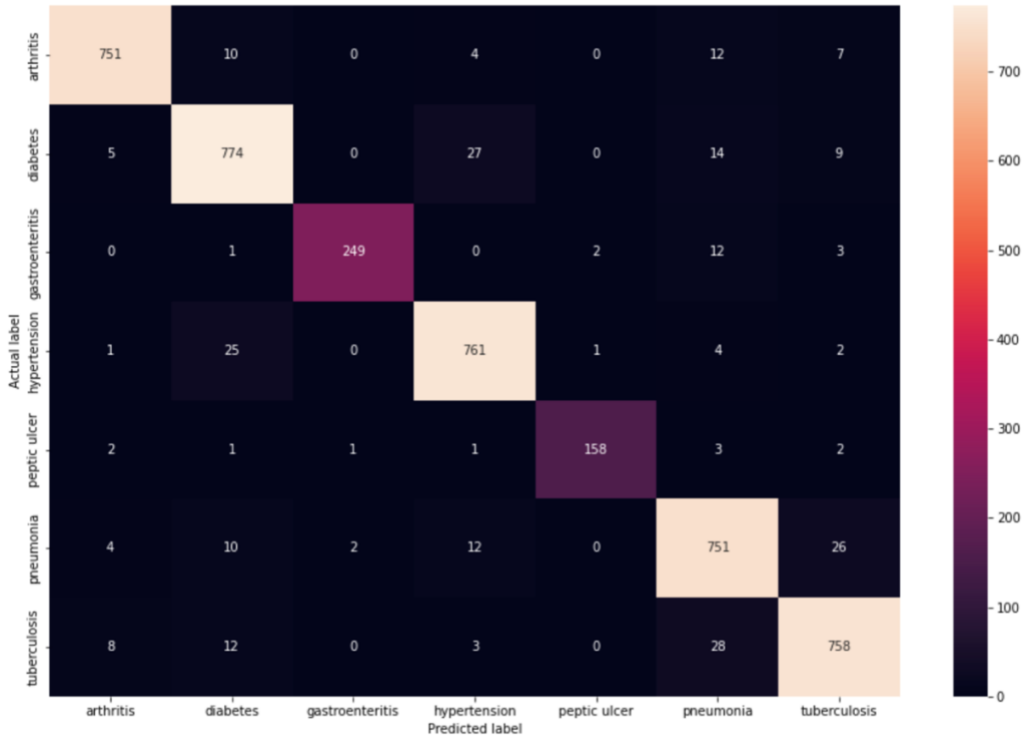

Figure 4: Classification Report and Confusion matrix for SVM Model

	precision	recall	f1-score	support
arthritis	0.98	0.95	0.96	784
diabetes	0.91	0.94	0.92	829
gastroenteritis	0.99	0.88	0.93	267
hypertension	0.95	0.96	0.95	794
peptic ulcer	0.98	0.88	0.93	168
pneumonia	0.89	0.93	0.91	805
tuberculosis	0.93	0.94	0.93	809
accuracy			0.94	4456
macro avg	0.95	0.92	0.93	4456
weighted avg	0.94	0.94	0.94	4456

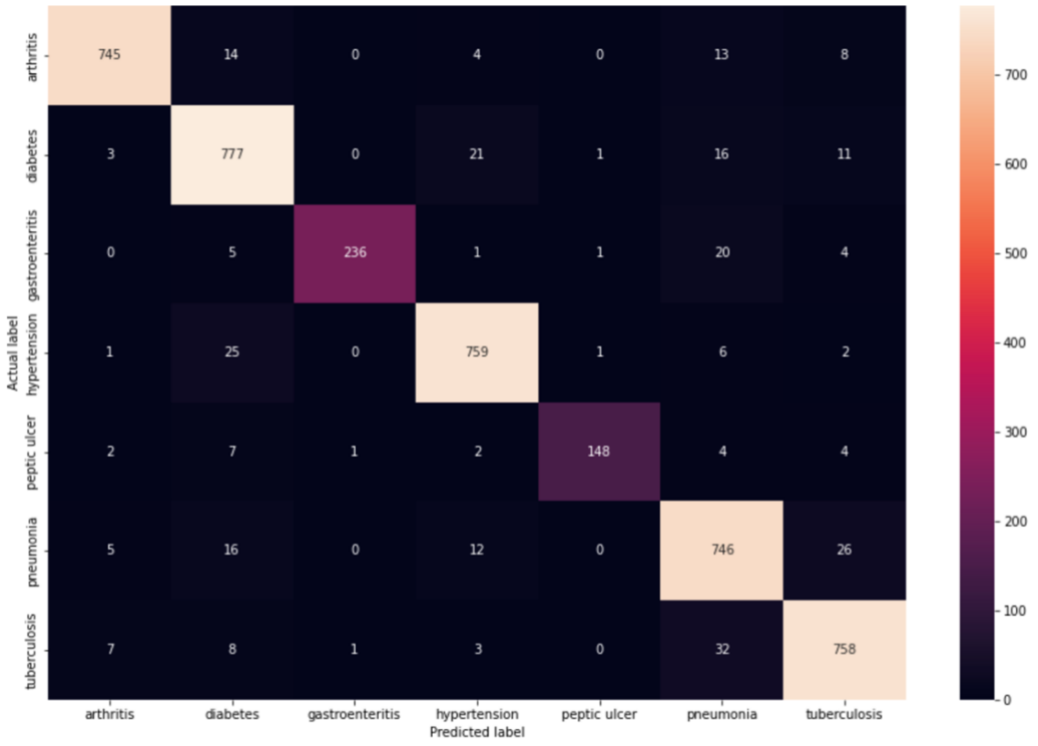

Figure 5: Classification Report and Confusion matrix for Linear Regression Model

	precision	recall	f1-score	support
arthritis	0.96	0.94	0.95	784
diabetes	0.90	0.93	0.91	829
gastroenteritis	0.99	0.85	0.92	267
hypertension	0.92	0.95	0.94	794
peptic ulcer	0.97	0.82	0.89	168
pneumonia	0.88	0.91	0.89	805
tuberculosis	0.91	0.92	0.91	809
accuracy			0.92	4456
macro avg	0.93	0.90	0.92	4456
weighted avg	0.92	0.92	0.92	4456

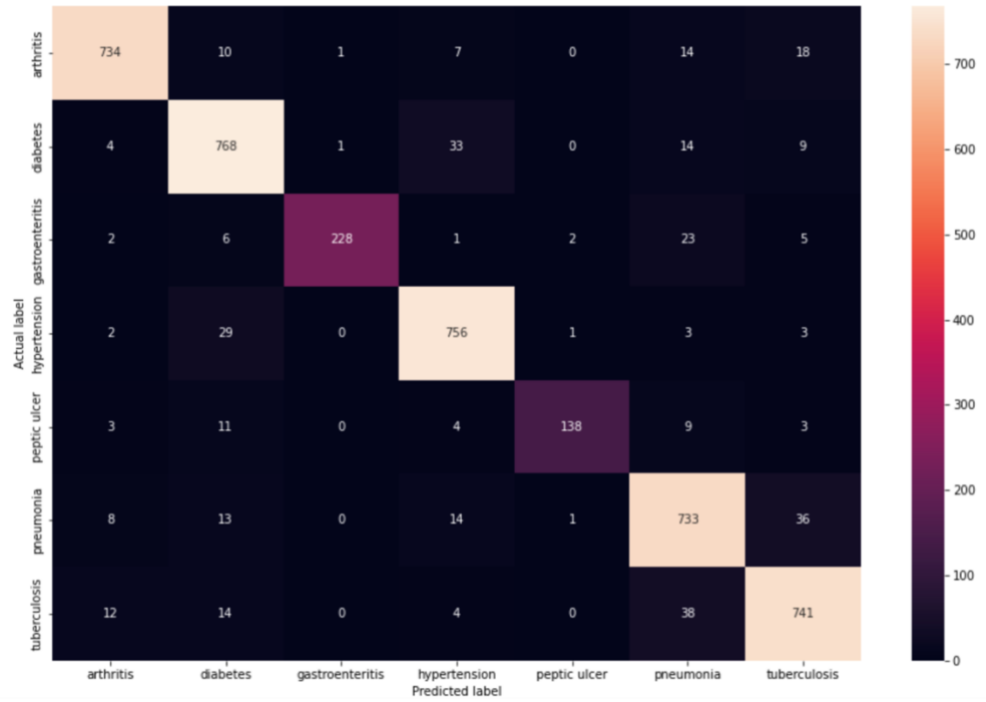

Figure 6: Classification Report and Confusion matrix for Random Forest Model

	precision	recall	f1-score	support
0	0.97	0.93	0.95	414
1	0.94	0.86	0.90	450
2	0.92	0.92	0.92	146
3	0.90	0.94	0.92	386
4	0.87	0.80	0.83	75
5	0.89	0.89	0.89	383
6	0.92	0.90	0.91	374
micro avg	0.92	0.90	0.91	2228
macro avg	0.92	0.89	0.90	2228
weighted avg	0.92	0.90	0.91	2228
samples avg	0.90	0.90	0.90	2228

Test set
Loss: 0.386
Accuracy: 0.915

	precision	recall	f1-score	support
0	0.93	0.96	0.94	414
1	0.94	0.88	0.91	450
2	0.96	0.90	0.93	146
3	0.93	0.92	0.93	386
4	0.88	0.88	0.88	75
5	0.90	0.89	0.90	383
6	0.90	0.91	0.91	374
micro avg	0.92	0.91	0.92	2228
macro avg	0.92	0.91	0.91	2228
weighted avg	0.92	0.91	0.92	2228
samples avg	0.91	0.91	0.91	2228

Figure 7: Classification Report for CNN(Top) and BRNN(Bottom)

Comparing two neural networks, we saw that the CNN performed better than the BRNN. From initial research about these two types of networks we found that that BRNN works best on sequential text. Since our text is about classification rather than predicting the next word based on the previous and future words, we believe the BRNN could not perform well on this dataset.

The 1D CNN is suited for text classification and hence performed better.

```
Epoch 1/10
282/282 [==============================] - 483s 2s/step - loss: 1.0184 - accuracy: 0.6039 - val_loss: 0.5083 - val_ac
curacy: 0.8444
Epoch 2/10
282/282 [==============================] - 489s 2s/step - loss: 0.3902 - accuracy: 0.8808 - val_loss: 0.4404 - val_ac
curacy: 0.8698
Epoch 3/10
282/282 [==============================] - 475s 2s/step - loss: 0.3159 - accuracy: 0.9067 - val_loss: 0.3142 - val_ac
curacy: 0.9107
Epoch 4/10
282/282 [==============================] - 483s 2s/step - loss: 0.2073 - accuracy: 0.9399 - val_loss: 0.2790 - val_ac
curacy: 0.9207
Epoch 5/10
282/282 [==============================] - 477s 2s/step - loss: 0.1898 - accuracy: 0.9452 - val_loss: 0.3759 - val_ac
curacy: 0.9042
Epoch 6/10
282/282 [==============================] - 476s 2s/step - loss: 0.1631 - accuracy: 0.9529 - val_loss: 0.2894 - val_ac
curacy: 0.9217
Epoch 7/10
282/282 [==============================] - 481s 2s/step - loss: 0.1408 - accuracy: 0.9577 - val_loss: 0.2801 - val_ac
curacy: 0.9202
70/70 [==============================] - 24s 337ms/step - loss: 0.3101 - accuracy: 0.9093
Test set
  Loss: 0.310
  Accuracy: 0.909
```

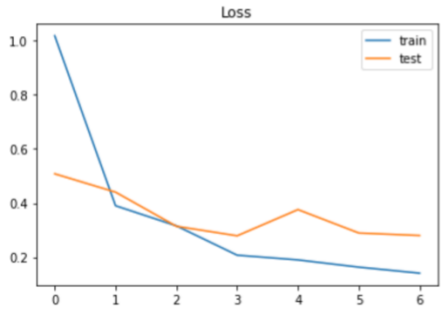

Figure 8: Classification Report for Convolutional Neural Network

```
Epoch 1/10
282/282 [==============================] - 12s 43ms/step - loss: 0.7117 - accuracy: 0.7405 - val_loss: 0.3331 - val_a
ccuracy: 0.9132
Epoch 2/10
282/282 [==============================] - 12s 42ms/step - loss: 0.2052 - accuracy: 0.9409 - val_loss: 0.2992 - val_a
ccuracy: 0.9152
Epoch 3/10
282/282 [==============================] - 12s 43ms/step - loss: 0.1196 - accuracy: 0.9646 - val_loss: 0.3062 - val_a
ccuracy: 0.9162
Epoch 4/10
282/282 [==============================] - 12s 43ms/step - loss: 0.0786 - accuracy: 0.9765 - val_loss: 0.3367 - val_a
ccuracy: 0.9232
Epoch 5/10
282/282 [==============================] - 12s 44ms/step - loss: 0.0603 - accuracy: 0.9809 - val_loss: 0.3665 - val_a
ccuracy: 0.9247
70/70 [==============================] - 1s 8ms/step - loss: 0.3862 - accuracy: 0.9147
Test set
  Loss: 0.386
  Accuracy: 0.915
```

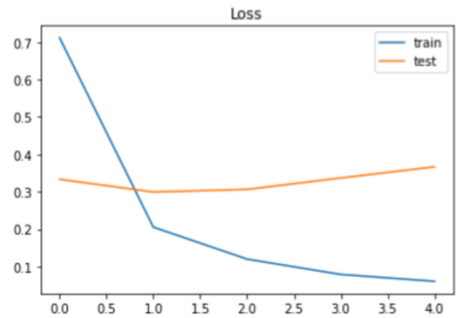

Figure 9: Classification Report for Convolutional Neural Network

Even though the accuracy of CNN is better than BRNN we can see the loss function for the test set does not decrease with the epoch. The loss function decrement is better in BRNN.

Answers to the research questions

- Our research found that SVM is the best classification models for plain text classification.
- Our research also found that neural networks did better when longer phrases are provided as input.

Future research directions

While extracting abstracts and diseases from PubMed, most abstract also had the disease word in it. For example: if we extracted abstract related to tuberculosis, it had the word tuberculosis in it already. It becomes very easy for any model to predict it. Removing those words can prevent

overfitting and make sure that predicted value is accurate in case the word is not present. We did this, and we saw slight improvement (reduction in high accuracy values)

In the future, we can add more diseases to our list. Currently, our model works for 7 diseases only. We can also increase the size of the dataset per disease which could help the models increase their accuracy. We have limited our dataset to contain the abstracts of the medical journal for our initial model. Adding more sources like introduction rather than just the abstract of a paper or medical blog article could be something we can investigate in the future.

As our initial goal was to classify disease based on doctor-patient conversation, one of our future works would be to collect or create such data and test it on the current model. There is a potential to create an end-to-end model, where the doctor-patient conversation is transcribed by a transcription model and is fed into our classification model. The end-to-end model would give the probable disease based on real-time conversation between the doctor and the patient.

The current neural networks have parameters based on a few manual tests. Automating these hyperparameters could be another way to increase the performance of neural networks. Cross-validation of the neural networks remains a to-do in our project.

Team members roles

1) Milestone One: Data Collection/Cleaning (**COMPLETED**)
    a) Abstract extraction from PyMed – Ruben/Shirish
    b) Data Cleaning- Aakriti/Shirish
    c) Feature Extraction/Selection - Aakriti
2) Milestone Two: Classification Model Implementation **(COMPLETED)**
    a) Classification Models Implementation – Shirish/Ruben
    b) Test with available dataset – Shirish/Ruben

c) Model Evaluation – Aakriti

3) Milestone Three: Documentation and Presentation **(COMPLETED)**

a) Presentation – Aakriti/Ruben/Shirish

b) Documentation - Aakriti/Ruben/Shirish

Things we learned from the project

- Data extraction from PubMed: Using python script to extract data from PubMed medical journal
- Natural Language Processing: The data cleaning process, text tokenization and lemmatization.
- Feature Extraction/TF-IDF: Using TF-IDF we can emphasize important words and ignore frequently occurring words that appear in all other dataset.
- Model Comparison: Since we used many models, we compared various models.
- Neural Networks (BRNN and CNN): We implemented two neural networks. We learnt that neural networks have a lot of hyperparameters.

Conclusion

It is possible to extract large plain text datasets and get meaningful insight from them. We can also emphasize specific words while creating a model using TF-IDF. This can make our program time-efficient and accurate while predicting. Different classification models were used for our disease diagnosis problem and SVM (Support Vector Machine) gave the highest accuracy.

Bibliography

Keniya, R., Khakharia, A., Shah, V., Gada, V., Manjalkar, R., Thaker, T., . . . Mehendale, N. (2020). Disease prediction from various symptoms using machine learning. *SSRN*.

Li, S. (2019, April 10). *Multi-Class Text Classification with LSTM*. Retrieved from Towards Data Science: https://towardsdatascience.com/multi-class-text-classification-with-lstm-1590bee1bd17

Shaikh, J. (2017, July). *Machine Learning, NLP: Text Classification using scikit-learn, python and NLTK.* Retrieved from TowardsDataScience: https://towardsdatascience.com/machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a

Shortliffe, E. (n.d.). Retrieved from nist.gov: https://www.nist.gov/system/files/documents/healthcare/NIST-Siegel_PART-3.pdf

Tensorflow. (n.d.). *Text classification with an RNN*. Retrieved from Tensorflow: https://www.tensorflow.org/text/tutorials/text_classification_rnn#stack_two_or_more_lstm_layers