Ontology-aware Learning from Electronic Health Records

Dissertation submitted in partial fulfilment for the award of the degree

Master of Technology in Computer Science

by

NIDHI PAL

Roll No.: CS2032
M.Tech, 2nd year

Under the supervision of
Dr. Malay Bhattacharyya

Computer and Communication Sciences Devision
INDIAN STATISTICAL INSTITUTE

July, 2022

CERTIFICATE

This is to certify that the work presented in this dissertation titled "Ontology-aware Learning from Electronic Health Records", submitted by Nidhi Pal, having the roll number CS2032, has been carried out under my supervision in partial fulfilment for the award of the degree of Master of Technology in Computer Science during the session 2021-22 in the Computer and Communication Sciences Division, Indian Statistical Institute. The contents of this dissertation, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Malay Bhattacharyya
Assistant Professor, Machine Intelligence Unit
Associate Member, Centre for Artificial Intelligence and Machine Learning
Associate Member, Technology Innovation Hub on Data Science, Big Data Analytics, and Data Curation
Indian Statistical Institute, Kolkata

Acknowledgements

First and foremost, I take this opportunity to express my sincere thankfulness and deep regard to *Dr. Malay Bhattacharyya*, for the impeccable guidance, nurturing and constant encouragement that he had provided me during my post-graduate studies. Words seem insufficient to utter my gratitude to him for his supervision in my dissertation work. Working under him was an extremely knowledgeable experience for a young researcher like me.

I also thank the CSSC and ISI Library for extending their supports in may different ways in my urgent need.

I shall forever remain indebted to my parents, teachers and friends for supporting me at every stage of my life. It is their constant encouragement and support that has helped me throughout my academic career and especially during the research work carried out in the last one year.

Date: 03-07-2022

Nidhi Pal
Roll No.: CS2032
M.Tech, 2nd year
Indian Statistical Institute

Abstract

Advances in natural language processing (NLP) in recent times has shown a great promise in improving the patient profiles with the help of their clinical notes. In medical practices, preparing clinical details for patients often happen through longer forms, which are really difficult to maintain and process. Therefore, people use abbreviations (writing a medical term in a shorter form) to record clinical details. In clinical notes, abbreviations are used recklessly without mentioning their definitions. These abbreviations can have different expansions based on their medical context. For example, the abbreviation "ivf" may denote either "intravenous fluid" or "in vitro fertilization" based on their contexts. It is thus a challenging task for NLP systems to correctly disambiguate abbreviations in their clinical notes. We have used the Naive Bayes approach for correctly disambiguating medical concepts and abbreviations by using NLP models. We have proposed a measure to find whether a given medical abbreviation is related to COVID or non-COVID. We have trained our model on the COVID ontologies and general medical concepts and tested it on the dataset which we have compiled at our own. We have tried to determine the correct senses for an abbreviation based on the associated context.

Contents

List of Figures

1 Introduction

In medical terminology creating clinical records for the patient profile is quite an important task. During the process of preparing patient profile, doctors often faces the issue of long medical terms which is difficult to pronounce, remember and understand. so, researchers used the abbreviations for the long forms convert into short forms, for make it easier to understand. Problem of correctly disambiguate the abbreviation corresponding to the given sentence is difficult.

from past many years various approaches have been used for resolve the problem. in this work, we have done some work on finding COVID related concepts for the given abbreviation with the help of the score, based on the occurrence of $(COVID, non-COVID)$ words in a sentence. it may causes major problem while not correctly classify the abbreviation as we have used some abbreviation rt in a sentence of clinical records, it will be difficult to distinguish the expansion, there will be multiple expansion corresponding to the abbreviation as "respiratory therapy" or "radiation therapy". here, both the definitions are related to the same problem, both of the expansions are lungs related so it will be difficult to identify the correct expansion of abbreviation. we have used Laplacian Naive Bayes and log probability in this approach for making calculation easier. abbreviation Detection in unrestricted text is a challenging task. In the area of general English, different methods have been developed for detecting the abbreviation in unrestricted text.

2 Problem Definition

Our main problem is to distinguish the medical abbreviations that might have different expansions. We have particularly focused on COVID based medical terms in this thesis. Our goal is twofold as listed below.

- We have identified whether a given concept is COVID related or not with the help of Multinomial Naive Bayes algorithm
- For the given sentence that contains a medical abbreviation, we have explored whether the abbreviation is COVID related or not by using the log of probabilities of the supporting words.

We have been given a list of abbreviations and its possible expansions, their type (COVID or non-COVID), and a corresponding sentence. The output will be the suggested expansion. A schematic diagram highlighting the problem is shown in Fig. 1.

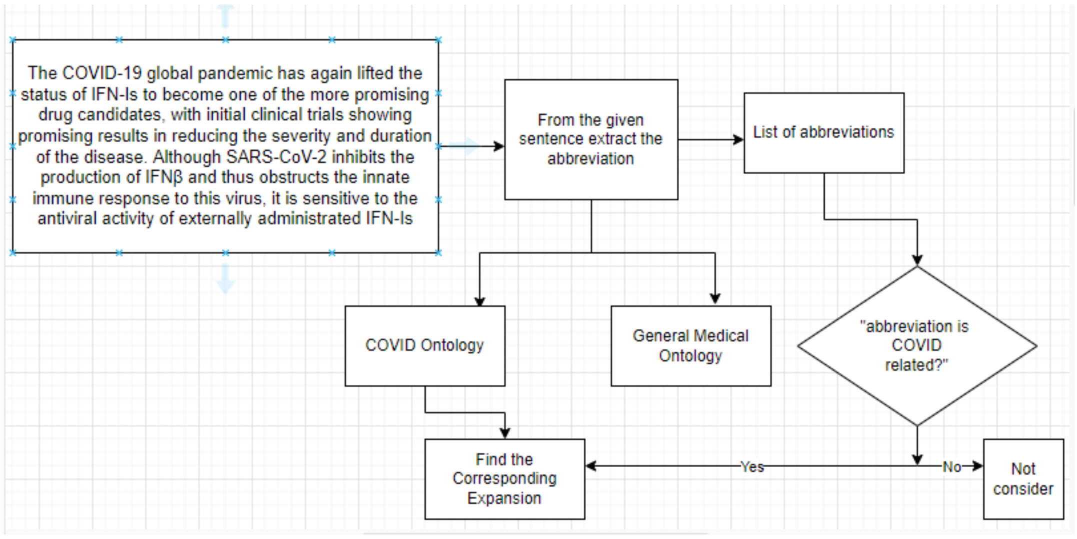

The COVID-19 global pandemic has again lifted the status of IFN-Is to become one of the more promising drug candidates, with initial clinical trials showing promising results in reducing the severity and duration of the disease. Although SARS-CoV-2 inhibits the production of IFNβ and thus obstructs the innate immune response to this virus, it is sensitive to the antiviral activity of externally administrated IFN-Is
From the given sentence extract the abbreviation
List of abbreviations
COVID Ontology
General Medical Ontology
"abbreviation is COVID related?"
Find the Corresponding Expansion
Yes
No
Not consider

Figure 1: A schematic diagram of the problem.

3 Related Work

In biomedical terminology, sometimes abbreviations occur together with their expanded forms at least once in the document. Usually, it appears in this format in short forms(long forms). for that several approaches have been developed to map the abbreviation with their correct expansions. which had been used to develop a database, this database is called sense inventory [8]. which contains the abbreviations with their detected possible senses.

There are many earlier research attempts in distinguishing between the ambiguous biomedical terms. Some of these are focused toward using examples generated from the UMLS Metathesaurus [10]. However, the work on expanding the disambiguous medical abbreviations is a relatively new domain. The expansion of abbreviated medical terms is an essential problem in medical science, precisely in Prescriptomics. Prescriptomics deals with the data collected from digitized prescriptions.

There have been several methods introduced for medical abbreviation disambiguation in recent times. This mainly includes supervised learning with decision tree, classification, Naive Bayes, SVM models, etc. Due to the availability of limited datasets, several problems arise for those models. So earlier research has attempted to create hand-labeled datasets for the said purpose. Former studies have used some hand-labeled datasets such as i2b2 hand-labeled data which contain 250 abbreviations [9], and some information about the patient. Recent researchers have also used publicly available datasets like CASI, mimic-3, UMLS, i2b2, all acronyms, and ADAM. CASI dataset contains the abbreviation and patient clinical records. Previously for distinguishing medical abbreviations, used the approach of reverse substitution. In this, if the data includes some long forms replace them with the short forms and write the long forms into the training dataset and the corresponding short forms as abbreviations. Skreta et al. have used the closely related medical concept not present in data [9]. Then they augmented it with the data.

Contextualized embedding has also been used for medical abbreviations. with the help of TF-IDF score also used for disambiguating medical abbreviations and the approach of local context used. in this approach find the correct expansion corresponding to the given abbreviation based on the context. in this case, the problem is if there is an abbreviation as rt and having the expansion as radiation therapy or respiratory therapy both of the words are lungs related. Therefore, it will be difficult to differentiate the expansion for the abbreviation based on the local context, local context contains information about the sentence only. So, they have used the approach, to differentiate medical abbreviations based on both global context and local context, global context contains all the information about the whole document.

In biomedical terminology, there are various knowledgeable sources available that may help to detect the abbreviation and their definitions. Sources have contains the abbreviation and its possible senses[12]. UMLS[2](Unified Medical Language System), CUI(Concept Unique Identifiers), ADAM[14](another database of abbreviations in MEDLINE), MEDLINE[6], and All Acronyms[5] are the available sources. The UMLS combines many biomedical vocabularies including the clinical text. The Metathesaurus file "MRCONSO.RRF" of UMLS contains information which is associated with medical terms, such as concept unique identifier. CUI is also used as the source of medical terms and abbreviations. ADAM contains the acronym and non-acronym abbreviations and medical concepts.

Different definitions of the abbreviations extracted from MEDLINE and abstract based on the short form / long form methods. In the domain of clinical records, Berman has used the abbreviations from pathology reports and classified them based on the relationship between the abbreviations and their expansions which is useful for the implementation of abbreviations detection and for expansion methods.

In this work, we have used the multinomial naive Bayes model to find whether the sentence is COVID related or non-COVID related. then find whether the given abbreviation will be COVID-related or not. if it is COVID-related then assign the corresponding COVID-related expansion to the abbreviation. we have used the Laplacian Bayes to prevent the problem of zero probability.

4 DataSet Details

4.1 Dataset Collection

To evaluate our work we have used COVID ontology[3] and general medical ontology, we have used COVID ontology For the COVID dataset, and it contains 16,000 rows in which it contains concept id, table, and concepts. where concept id contained the identification number, concepts contained some sentences and table id contained some condition(situation). From this dataset, we only need concepts for our work.

We have dropped all the columns which are not required for our work. so, we have dropped concept id and table. at last, we have concepts of the COVID dataset which we have used in our work. for general medical data, we have to use the CASI dataset[4] as non-COVID data. which contains 21,000 rows of abbreviation, expansion, and sentences. we have extracted sentences from the CASI dataset and used them for our work. we have manually created datasets of general medical concepts which contain 300 concepts. in this dataset, we have contained abbreviation, expansion, source to get the abbreviation, type for defining whether it is COVID related abbreviation, or non-COVID related and sentence. we have named this dataset abbreviation data. and used it in our work as test data.

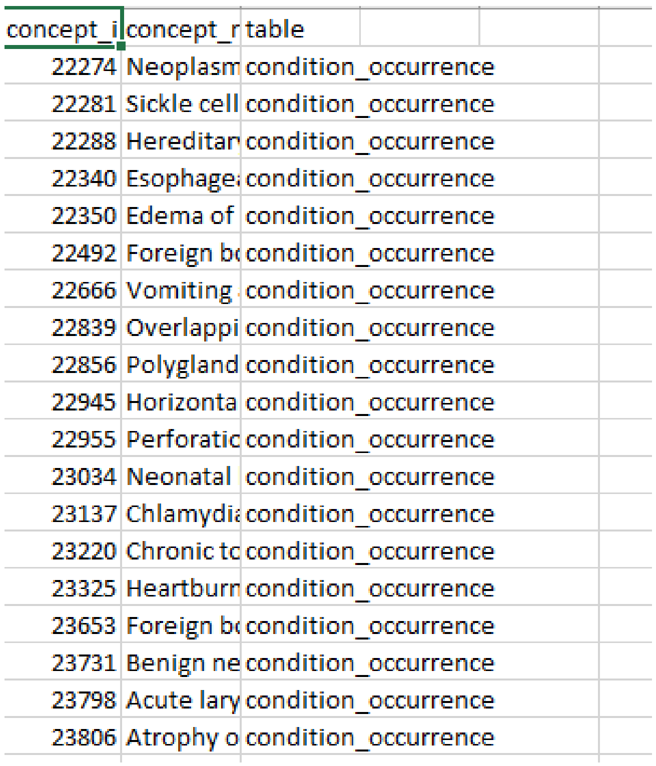
We have combined the COVID ontology and general medical datasets (non-COVID dataset), which contain 33,400 rows out of which 16,400 rows of COVID ontology and 17,000 rows of non-COVID data (we have picked these data from the CASI dataset). A snapshot of the data pertaining to COVID ontologies is shown in Fig. 2. Similarly, a snapshot of the data related to non-COVID that is taken from the CASI dataset is shown in Fig. 3.

This data represents the structure of the CASI dataset. we have used this data after performing data cleaning and pre-processing.

Now, the data which we have created as the test data and named it abbreviation data.

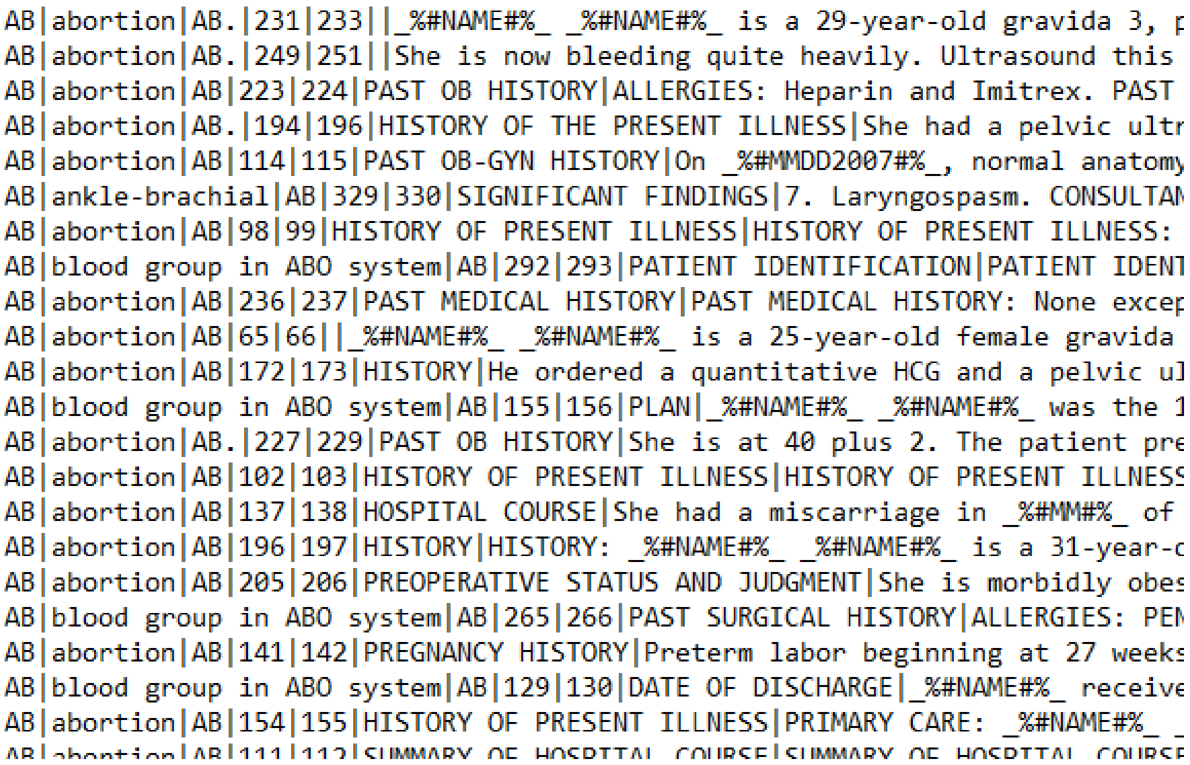
4.2 Data Pre-processing

After collecting all the required data we have to perform data cleaning and pre-processing of the data. we have dropped those columns from the data which is not required for our work. and removed all the rows from a dataset that is not present in the desired format. We have used Python libraries like pandas, NumPy, sklearn, matplotlib, word cloud, nltk, and text blob.



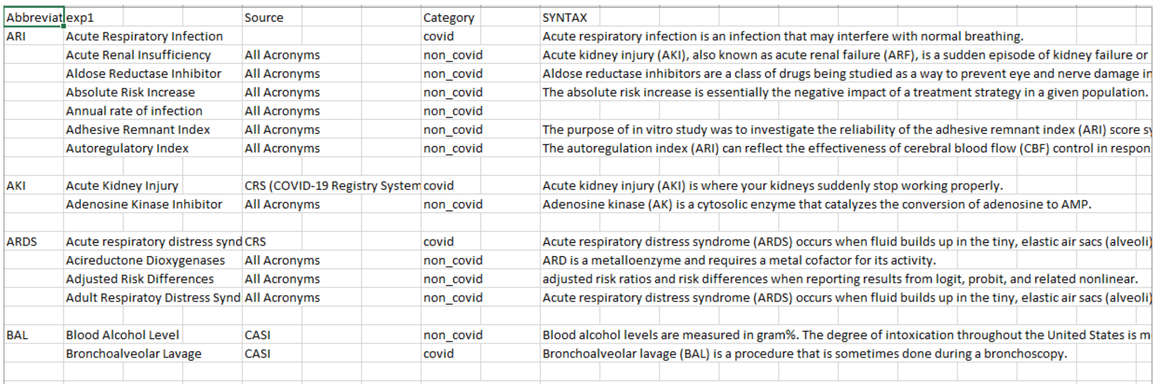
concept_i	concept_r	table
22274	Neoplasm	condition_occurrence
22281	Sickle cell	condition_occurrence
22288	Hereditar	condition_occurrence
22340	Esophage	condition_occurrence
22350	Edema of	condition_occurrence
22492	Foreign b	condition_occurrence
22666	Vomiting	condition_occurrence
22839	Overlappi	condition_occurrence
22856	Polygland	condition_occurrence
22945	Horizonta	condition_occurrence
22955	Perforatic	condition_occurrence
23034	Neonatal	condition_occurrence
23137	Chlamydi	condition_occurrence
23220	Chronic tc	condition_occurrence
23325	Heartburn	condition_occurrence
23653	Foreign b	condition_occurrence
23731	Benign ne	condition_occurrence
23798	Acute lary	condition_occurrence
23806	Atrophy o	condition_occurrence

Figure 2: Snapshot of data present in COVID ontologies.



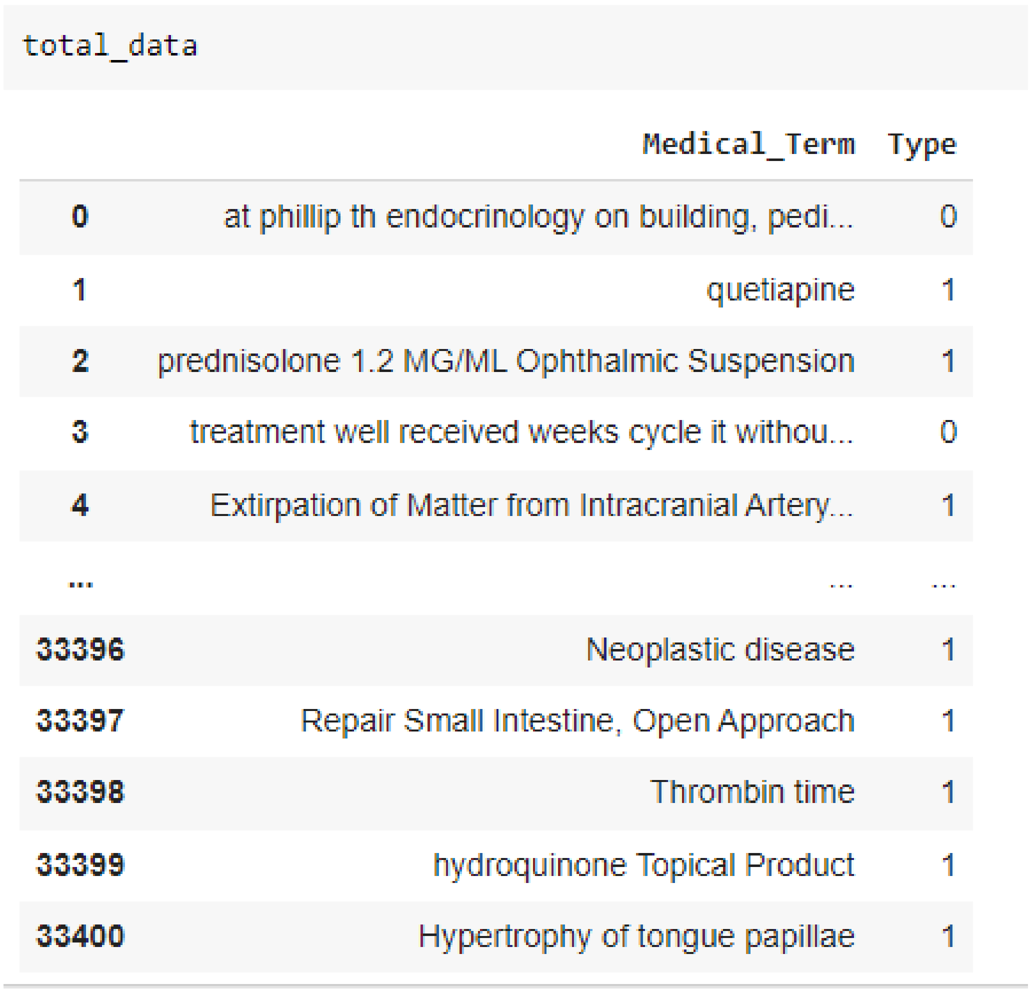
```
AB|abortion|AB.|231|233||_%#NAME#%_ _%#NAME#%_ is a 29-year-old gravida 3, p
AB|abortion|AB.|249|251||She is now bleeding quite heavily. Ultrasound this
AB|abortion|AB|223|224|PAST OB HISTORY|ALLERGIES: Heparin and Imitrex. PAST
AB|abortion|AB.|194|196|HISTORY OF THE PRESENT ILLNESS|She had a pelvic ultr
AB|abortion|AB|114|115|PAST OB-GYN HISTORY|On _%#MMDD2007#%_, normal anatomy
AB|ankle-brachial|AB|329|330|SIGNIFICANT FINDINGS|7. Laryngospasm. CONSULTAN
AB|abortion|AB|98|99|HISTORY OF PRESENT ILLNESS|HISTORY OF PRESENT ILLNESS:
AB|blood group in ABO system|AB|292|293|PATIENT IDENTIFICATION|PATIENT IDENT
AB|abortion|AB|236|237|PAST MEDICAL HISTORY|PAST MEDICAL HISTORY: None excep
AB|abortion|AB|65|66||_%#NAME#%_ _%#NAME#%_ is a 25-year-old female gravida
AB|abortion|AB|172|173|HISTORY|He ordered a quantitative HCG and a pelvic ul
AB|blood group in ABO system|AB|155|156|PLAN|_%#NAME#%_ _%#NAME#%_ was the 1
AB|abortion|AB.|227|229|PAST OB HISTORY|She is at 40 plus 2. The patient pre
AB|abortion|AB|102|103|HISTORY OF PRESENT ILLNESS|HISTORY OF PRESENT ILLNESS
AB|abortion|AB|137|138|HOSPITAL COURSE|She had a miscarriage in _%#MM#%_ of
AB|abortion|AB|196|197|HISTORY|HISTORY: _%#NAME#%_ _%#NAME#%_ is a 31-year-c
AB|abortion|AB|205|206|PREOPERATIVE STATUS AND JUDGMENT|She is morbidly obes
AB|blood group in ABO system|AB|265|266|PAST SURGICAL HISTORY|ALLERGIES: PEN
AB|abortion|AB|141|142|PREGNANCY HISTORY|Preterm labor beginning at 27 weeks
AB|blood group in ABO system|AB|129|130|DATE OF DISCHARGE|_%#NAME#%_ receive
AB|abortion|AB|154|155|HISTORY OF PRESENT ILLNESS|PRIMARY CARE: _%#NAME#%_ _
AB|abortion|AB|111|112|SUMMARY OF HOSPITAL COURSE|SUMMARY OF HOSPITAL COURSE
```

Figure 3: The non-COVID data collected from the CASI dataset.



Abbreviat	exp1	Source	Category	SYNTAX
ARI	Acute Respiratory Infection		covid	Acute respiratory infection is an infection that may interfere with normal breathing.
	Acute Renal Insufficiency	All Acronyms	non_covid	Acute kidney injury (AKI), also known as acute renal failure (ARF), is a sudden episode of kidney failure or
	Aldose Reductase Inhibitor	All Acronyms	non_covid	Aldose reductase inhibitors are a class of drugs being studied as a way to prevent eye and nerve damage in
	Absolute Risk Increase	All Acronyms	non_covid	The absolute risk increase is essentially the negative impact of a treatment strategy in a given population.
	Annual rate of infection	All Acronyms	non_covid	
	Adhesive Remnant Index	All Acronyms	non_covid	The purpose of in vitro study was to investigate the reliability of the adhesive remnant index (ARI) score sy
	Autoregulatory Index	All Acronyms	non_covid	The autoregulation index (ARI) can reflect the effectiveness of cerebral blood flow (CBF) control in respon
AKI	Acute Kidney Injury	CRS (COVID-19 Registry System	covid	Acute kidney injury (AKI) is where your kidneys suddenly stop working properly.
	Adenosine Kinase Inhibitor	All Acronyms	non_covid	Adenosine kinase (AK) is a cytosolic enzyme that catalyzes the conversion of adenosine to AMP.
ARDS	Acute respiratory distress synd	CRS	covid	Acute respiratory distress syndrome (ARDS) occurs when fluid builds up in the tiny, elastic air sacs (alveoli)
	Acireductone Dioxygenases	All Acronyms	non_covid	ARD is a metalloenzyme and requires a metal cofactor for its activity.
	Adjusted Risk Differences	All Acronyms	non_covid	adjusted risk ratios and risk differences when reporting results from logit, probit, and related nonlinear.
	Adult Respiratoy Distress Synd	All Acronyms	non_covid	Acute respiratory distress syndrome (ARDS) occurs when fluid builds up in the tiny, elastic air sacs (alveoli)
BAL	Blood Alcohol Level	CASI	non_covid	Blood alcohol levels are measured in gram%. The degree of intoxication throughout the United States is m
	Bronchoalveolar Lavage	CASI	covid	Bronchoalveolar lavage (BAL) is a procedure that is sometimes done during a bronchoscopy.

Figure 4: Snapshot of manually created Test data .



```
total_data
```

	Medical_Term	Type
0	at phillip th endocrinology on building, pedi...	0
1	quetiapine	1
2	prednisolone 1.2 MG/ML Ophthalmic Suspension	1
3	treatment well received weeks cycle it withou...	0
4	Extirpation of Matter from Intracranial Artery...	1
...	...	...
33396	Neoplastic disease	1
33397	Repair Small Intestine, Open Approach	1
33398	Thrombin time	1
33399	hydroquinone Topical Product	1
33400	Hypertrophy of tongue papillae	1

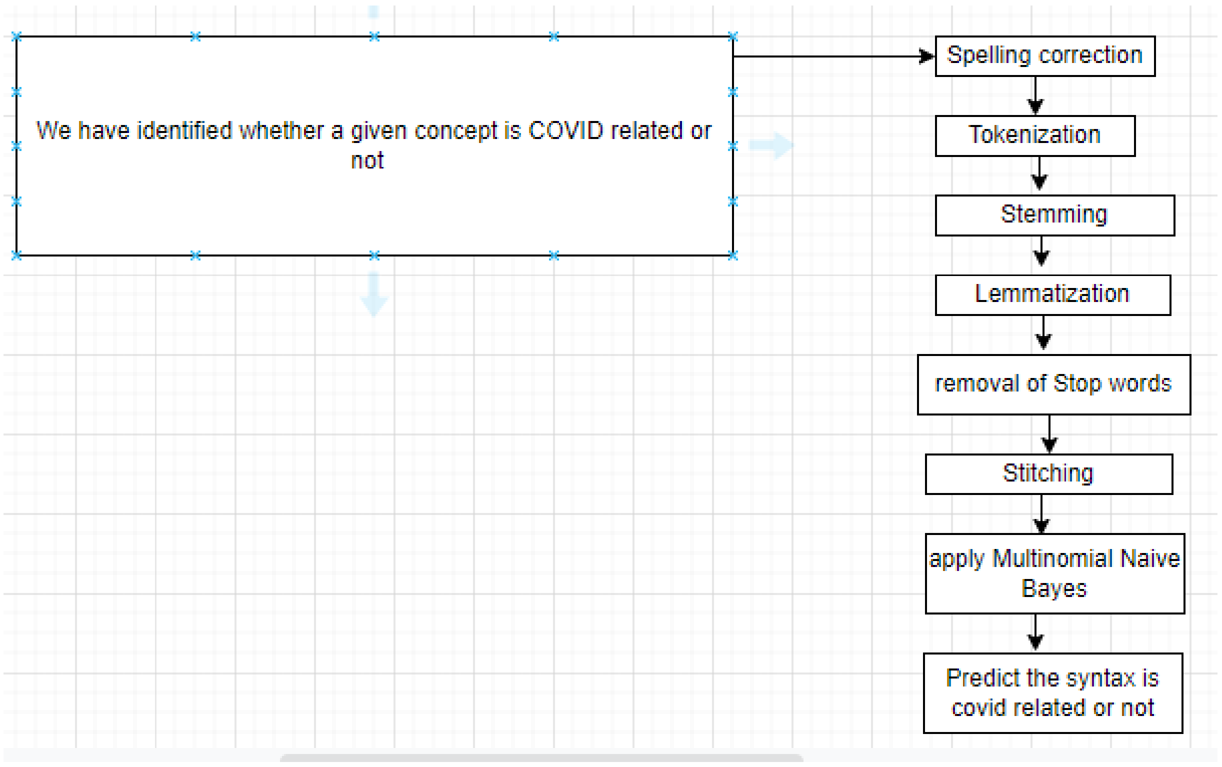
Figure 5: Snapshot of Total data .

This dataset represents the total dataset after combining COVID ontology and general medical data(non-COVID data). this data having index and medical term which contains concepts and type which defines that given sentence or concept is COVID related or non COVID related.for better understanding we have use .describe() function. for data it will provide the important details of data set, for better understanding of the data. this function will give the details as max, min, avg, count, mean, std, 25%, 50%, 75%. Because of the data combining indexes had been shuffled, to solve this problem we have used reset.index() function for re-indexing.

5 Distinguishing COVID and non-COVID Concepts

In this work, we have used the multinomial naive Bayes[1] approach. To find the Given sentence of the data, if it is COVID related or non-COVID related.

We have perform these steps here.



We have identified whether a given concept is COVID related or not
Spelling correction
Tokenization
Stemming
Lemmatization
removal of Stop words
Stitching
apply Multinomial Naive Bayes
Predict the syntax is covid related or not

Figure 6: steps for The Approach

We have created a dataset for COVID and non-COVID data. First of all, we have gathered all the required data for our work and applied data preprocessing. we have imported all the required libraries. In the next step read all the required datasets by using pd.read_csv() function. We have used concatenate function to concatenate the datasets. In which we have used $16,400$ rows of COVID ontology and $17,000$ rows of non-COVID data. and then assign the new combined datasets with the new term as total data with 33,400 rows. After combining the datasets we have to perform reindexing.

From the original format to making the datasets into our desired format we have to perform some NLP functions. first, we have to convert all the letters into lower case. and remove usernames, URLs, and all digits, all single characters, punctuation, and quotes, convert more than 2 letter repetitions to 2 letters and replaces double spaces with single space. we have renamed the columns, "concepts" as "medical term" and type as "category". we have used `dropna` function for remove unknown, NaN and null values.

We are using the Multinomial Naive Bayes approach for text classification. this is a task of natural language processing. Multinomial Naive Bayes is a supervised algorithm of classification. Here, we have used this approach in our work to classify whether a sentence is COVID-related or non-COVID-related. we have to find the probability of being COVID related for each word of the sentence.in Multinomial Naive Bayes we used Bayes Theorem. and same as we have found the probability of being non-COVID for each word. and we used the log probabilities to make the calculation easier. normalized the probability of COVID and the probability of non-COVID.

5.1 Spelling Correction

Spelling correction is a task of NLP. we have the data, in certain it can be noisy means it can have spelling errors, so we have to correct them. in NLP there are many algorithms defined for spelling correction corresponding to the problem. we have used spelling correction for incorrect spelling. in our work, we have not used spelling correction. but it can be used in the future. we have used blobber from text blob as a library and used NaiveBayesAnalyzer from textblob.sentiments. To use the spelling correction function.

5.2 Tokenization

We have used tokenization, and we know tokenization is a process of segmenting a string of characters into words. We use tokenization to find out tokens. Basically token is an instance of a sequence of characters.

5.3 Sentence Segmentation

Sentence segmentation we used for many purposes. Tokenization is one of them. Sentence segmentation is the problem of deciding where the sentence begins and where it ends.

5.4 Stemming

We have used stemming for reducing terms to their stem, which means chopping of suffices and crude chopping of affixes. It is language-dependent. For example, –automate, automatic, automation all are reduced into 'automat'. we used Porter's algorithm for stemming [11].

5.5 Lemmatization

We used lemmatization because we have try to find whether the actual dictionary had the given word with their form or not. So lemmatization reduces the inflections or variant forms into their root form.

5.6 Stop Words

We have used stop words here. Every language has its own stop words. we have used the English language here, so we have used the stop words of the English language.

After completing those steps we have to perform data cleaning. In which we removed punctuation, because of that we got data in form of alphabets. So we have used the .join function for joining these alphabets. after that, we used the .split function for forming the sentence into tokens or words. Then, we used a count vectorizer to convert every word into vectors based on the frequency that occurs in the sentence.

5.7 Summary of Pre-processed Data

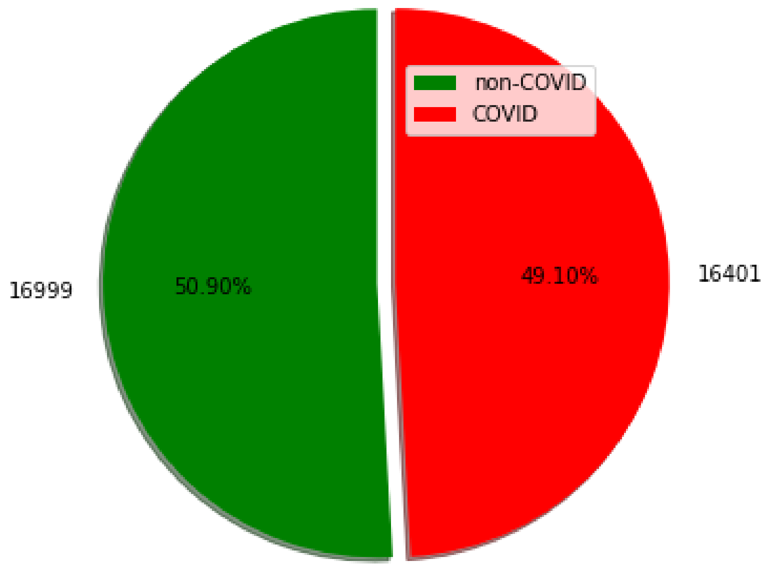


Figure 7: The share of COVID and non-COVID concepts in the data prepared for predictive analysis.

Fig. 7 represents 16,999 data points of non COVID data belongs to the green region. on the other hand 16,401 data points of COVID data belongs to the red region. that means 50% non-COVID data and 49% COVID data points are presented in our dataset.

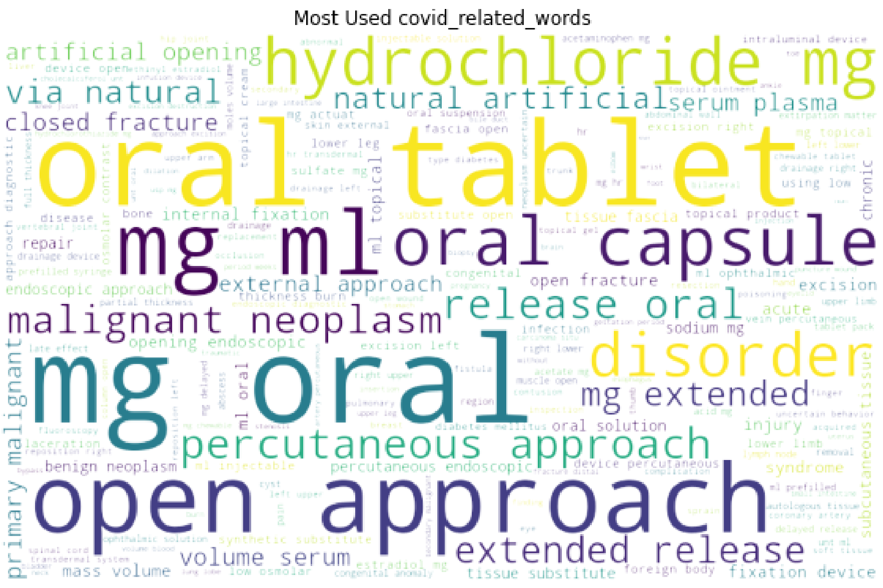


Figure 8: The most frequently used words in the COVID related data.

For better understanding of data we have plot the COVID and non-COVID data with the help of `matplotlib`.

We have also attempted to find out the most frequently occurring words in the COVID and non-COVID related data.

5.7.1 Visualizing the Word Could

Word cloud is used here, for a visual representation of words. Cloud creators in word cloud are used for highlighting the popular words (which is mostly used in our data) based on the frequency of words.

We have created both the word clouds for the COVID data (see) as well as non-COVID data.

5.7.2 plot for mostly used words for non COVID data.

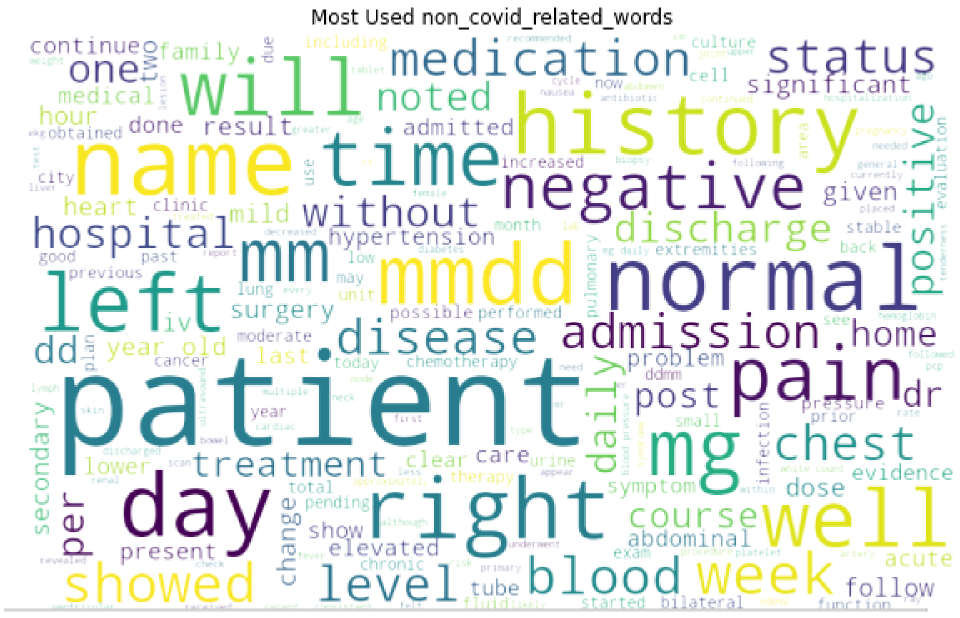


Figure 9: The most frequently used words in the non-COVID related data.

In our work, we have split data into 20% for the test data and 80 % for the training data. so, train data contains 26,720 rows and the columns X_train, y_train. for the test data contains 6,680 rows and columns X_test, y_test. after that, we applied the Multinomial Naive Bayes algorithm to the training data. so, here, we have learned the Naive Bayes classifier on X_train, y_train. we have find the predicted values on X_test.

5.8 Results

5.8.1 Performance Metrics

We have represented the accuracy with the help of confusion matrix. It contains a predicted label on the vertical axis and a true label on the horizontal axis, wherein the first quadrant shows the correctly determined non-COVID values and the last quadrant showed the correctly determined COVID values. The second and third quadrant shows the falsely predicted values. Thus, we have obtained an overall accuracy of 95%.

Based on this, we have calculated the values of precision, recall, f1−score, and support. We have used all these as performance metrics. Suppose we denote the True Positive, True Negative, False Positive, and False Negative as TP, TN, FP, and FN, respectively. Then values of precision and recall are calculated as follows.

$$\text{Precision} = \frac{TP}{TP+FP}$$
$$\text{Precision} = \frac{3165}{3165+162}$$
$$= 0.95$$

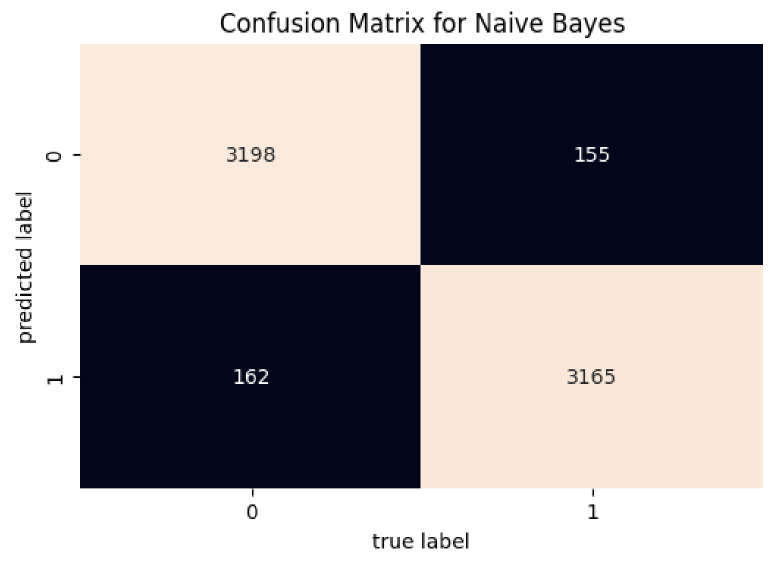
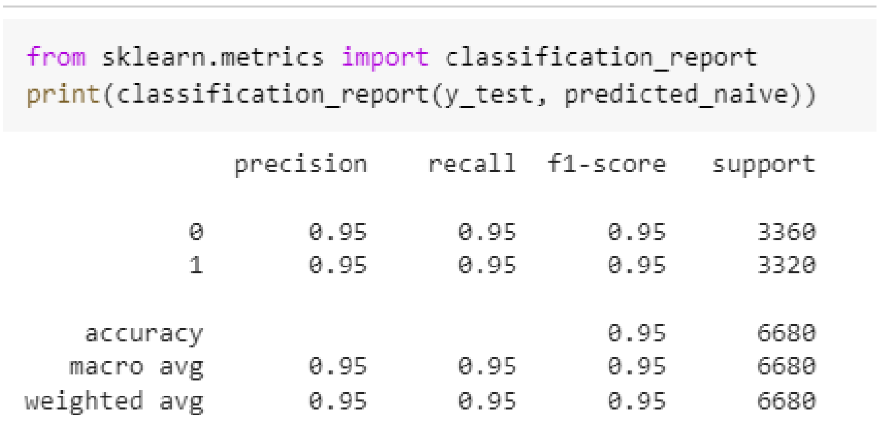


Figure 10: performance measurement based on the Confusion matrix

$$\text{Recall} = \frac{TP}{TP+FN}$$
$$\text{Recall} = \frac{3165}{3165+155}$$
$$= 0.95 \ \text{F1-score} = 2* \frac{Precision*Recall}{Precision+Recall} \ \text{F1-score} = 2* \frac{0.95*0.95}{0.95+0.95}$$
$$= 2* \frac{0.9025}{1.9}$$
$$= 0.95$$



```
from sklearn.metrics import classification_report
print(classification_report(y_test, predicted_naive))
```

```
              precision    recall  f1-score   support

           0       0.95      0.95      0.95      3360
           1       0.95      0.95      0.95      3320

    accuracy                           0.95      6680
   macro avg       0.95      0.95      0.95      6680
weighted avg       0.95      0.95      0.95      6680
```

Figure 11: The performance measurement based on the Precision and Recall.

We have also observed the performance with the help of AUC ROC.

The ROC plot reflects the True positive rate (when the truth is positive and the predicted value is also positive) on the vertical axis and the False positive rate (when the truth value is negative but the predicted value shows positive) on the horizontal axis.

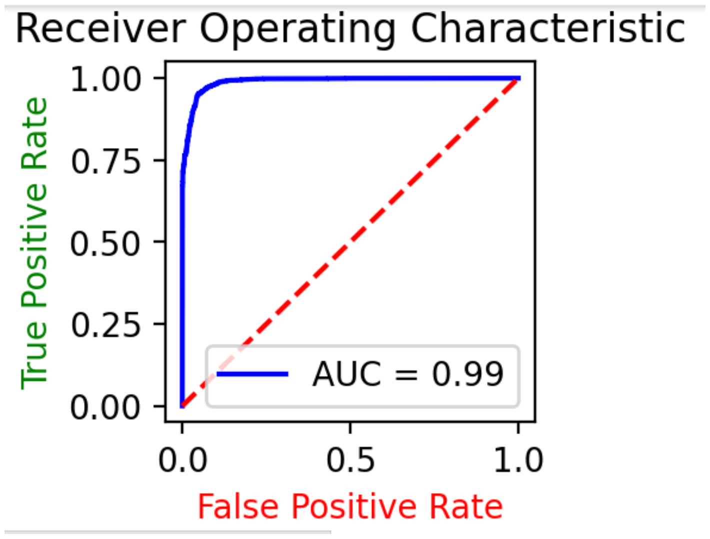
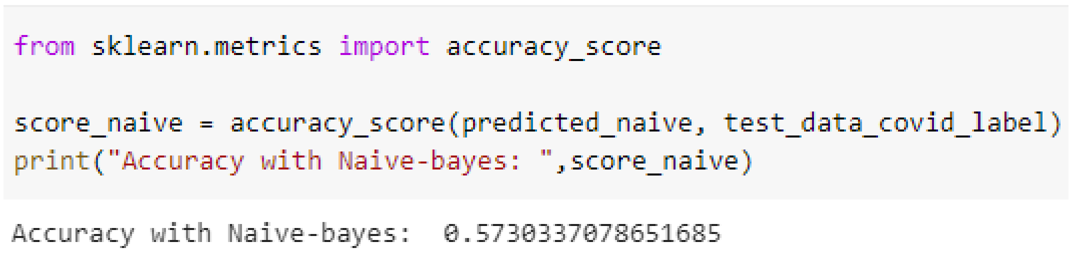


Figure 12: The performance measurement based on ROC.

Now, we used this model on the data which we have created as the test data (with abbreviation data) for the purpose of testing. We have read the data and dropped values. We dropped all the columns except the syntax from the test data and with the help of count vectorizer .transform it means we have used the same dimension as we used during learning the model and model_naive.predict function for predicting the values. We observed that the accuracy with the Naive Bayes classifier is 0.57.



```
from sklearn.metrics import accuracy_score

score_naive = accuracy_score(predicted_naive, test_data_covid_label)
print("Accuracy with Naive-bayes: ",score_naive)
```

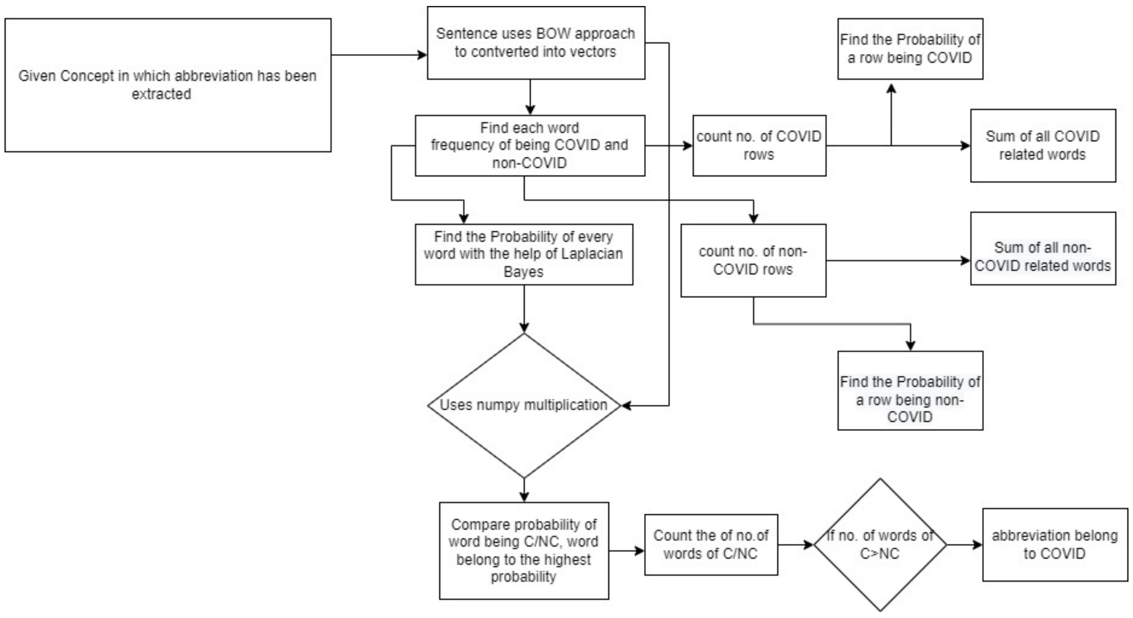
```
Accuracy with Naive-bayes:  0.5730337078651685
```

Figure 13: Accuracy with the Naive Bayes Approach

6 COVID Related Medical Term Disambiguation

Here, we have done some extra work. first, we have combined the data of COVID ontology and general medical ontology. we have picked 5000 rows of COVID ontology and 5000 rows of general medical ontology. we have performed data pre-processing and data cleaning, dropna values. merge all the data named as total data. and having 10,000 rows and 2 columns. as "Medical Terms" and "Type". "Type" contains the binary value '1' and '0' corresponding to COVID and non-COVID.

"For the given sentence that contains a medical abbreviation, we have explored whether the abbreviation is COVID related or not by using the log of probabilities of the supporting words. We have to perform these steps here".



Given Concept in which abbreviation has been extracted
Sentence uses BOW approach to contverted into vectors
Find each word frequency of being COVID and non-COVID
count no. of COVID rows
Find the Probability of a row being COVID
Sum of all COVID related words
Find the Probability of every word with the help of Laplacian Bayes
count no. of non-COVID rows
Sum of all non-COVID related words
Find the Probability of a row being non-COVID
Uses numpy multiplication
Compare probability of word being C/NC, word belong to the highest probability
Count the of no.of words of C/NC
If no. of words of C>NC
abbreviation belong to COVID

Figure 14: Flowchart for the entire Approach

we have used all the required libraries as numPy, pandas, matplotlib, seaborn, nltk, scikit. Data collection caused the problem of index shuffling, so we need to perform index resetting on the combined data by using the data reset function For converting the data into the desired format, we have used some functions. so in the first step, convert every word into lower case. Removed usernames, Remove URLs, Removes all digits, Remove (&quot), Removes all single characters, Removed all punctuation, Converted more than 2 letter repetitions to 2 letters, and Replaces double spaces with single spaces. these are the function we have used in our dataset.

We have to perform steps of spelling correction, Tokenization, Stemming, stop words, and lemmatization. these are the NLP steps we have used for receiving data into the desired format.

6.1 BOW Approach

We have used the BOW approach[13] to convert word into vectors, we have used the count vectorizer function. so we create a dictionary or corpus of all the distinct words from the data. corresponding to every sentence, if the word belongs to the sentence it will make it to 1 or use the frequency (how many times a word occur in a sentence) of that word. otherwise it will make it to 0 if the word is absent in a sentence.

We have split the data into train and test sets. We have used 20% of the data as the test set (having 2000 rows) and the rest 80% as train set (having 7,999 rows). training data contained columns as `X_train`, `y_train` and test data contained column as `X_test`, `y_test` in our representation.

Now, we have created the data frames for the train and test data as BOW Train and BOW Test. After that we have trained the model based on the X_train of BOW_Train. we have all the distinct words of data as columns and sentences of train data as rows. now, we have used BOW_Test. Here, we have used the .fit transform function, so the dimension will be the same as previous during the model training, with sentences of test data as rows. we have created a column named a category in BOW Train and BOW Test data. category value, column defines the status of sentences as it is COVID or non-COVID. the category values of BOW_Train used for model training and we have to predict the category value for BOW_Test.

We have grouped the rows based on the category where COVID is assigned by '1' and non-COVID is assigned by '0'. So, cons df contains the frequency for each word occurring in the rows of COVID data and non-COVID data, and category values that show '1' for 'COVID' and '0' for non-COVID. We have counted how many rows belong to a particular class. In our work, we have 4004 rows belonging to non-COVID and 3995 belonging to COVID. Thus, we have the count of all words related to non-COVID as 42275 and the count of all words related to COVID as 21605.

6.2 Laplace smoothing

We have used Laplace smoothing[7] to prevent our work from the problem of zero probability. For this, we have used a smoothing parameter alpha. Note that the value of alpha should not be equal to zero ($\alpha ! = 0$). By using this, the probability will never be zero irrespective of the fact whether a word is present or not in the data set. If we use higher values for alpha, it will push the likelihood toward the value of 0.5. So, we do not get any valuable information from this. Hence, we prefer to take the value of alpha as 1. The formula we are using for the Laplace Smoothing is as follows.

$$\text{P(word|being COVID related)} = \frac{number of sentences with word and y=COVID+\alpha}{Total no. of rows which is COVID related}$$

If we are taking some particular word as w which is present in sentences and it is given that the word w is COVID related denoted as c, assume, the number of sentences with w as w'. Number of COVID-related sentences as r then, then the probability of $P(w|c)$ will be as follows.

$$\text{P(w|c)} = \frac{w' and y = COVID + \alpha}{r}$$

We have created a data frame named Prob Table with the column P_c which contains the probability of COVID-related rows and non-COVID-related rows. The category count contains the number of COVID-related rows and the number of non-COVID-related rows presented.

$$\text{P(COVID related rows)} = \frac{no.of COVIDrelatedrows}{totalno.of rows}$$

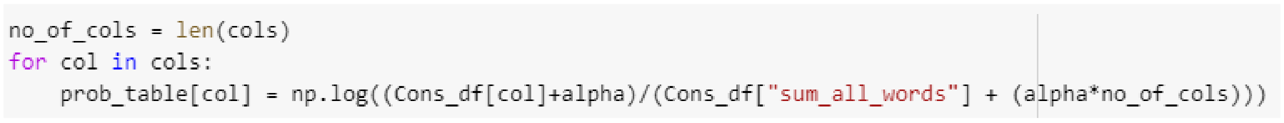
In our data, there are 4004 non-COVID-related rows where 3995 COVID related rows so P(COVID-related rows) will be as follows.

$$\text{P(COVID related rows)} = \frac{3995}{4004+3995} = 0.49943$$

Probability of total number of non-COVID related rows is calculated as.

$$\text{P(non-COVID related rows)} = \frac{4004}{4004+3995} = 0.500563$$

cols contain all the distinct words of the data. We have added a column col to the Prob Table which contained the probability for each COVID-related word with Laplacian smoothing

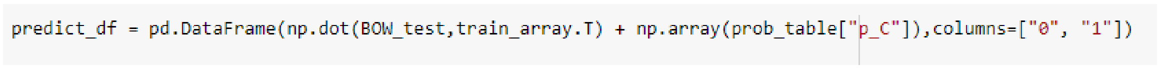


```
no_of_cols = len(cols)
for col in cols:
    prob_table[col] = np.log((Cons_df[col]+alpha)/(Cons_df["sum_all_words"] + (alpha*no_of_cols)))
```

Now, Prob Table has the columns 'P_c' (probability of total no.of COVID-related rows and non-COVID-related rows), 'category values'('1' for 'COVID' and '0' for non-COVID), and all the distinct word of the data. we have found the log probability for P_c of Prob Table. We have dropped the columns 'P_c' and 'category values' from them and assigned them to the cols. Now, We have to calculate the probability of every word.

Prob Table contained all the distinct words of data as columns, categories 'COVID' and 'non-COVID' as rows. We have to calculate the probability of every word of the given sentence, Consider the multiplication of the probability of all the words in a sentence.

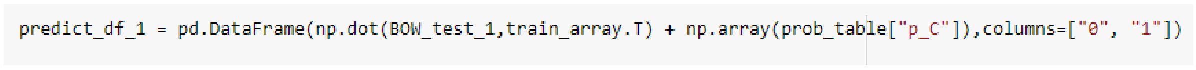
We have created a train array that contains all the words with their probability of COVID and non-COVID. now convert train array into numPy array. We predicted the probability for the test data(BOW Test) and store it into a data frame named predicted. first, take the transpose of the dot product of the BOW Test and train array, and sum it with the probability of that sentence from P_c of Prob Table. we have predicted the probability of BOW_Test.



```
predict_df = pd.DataFrame(np.dot(BOW_test,train_array.T) + np.array(prob_table["p_C"]),columns=["0", "1"])
```

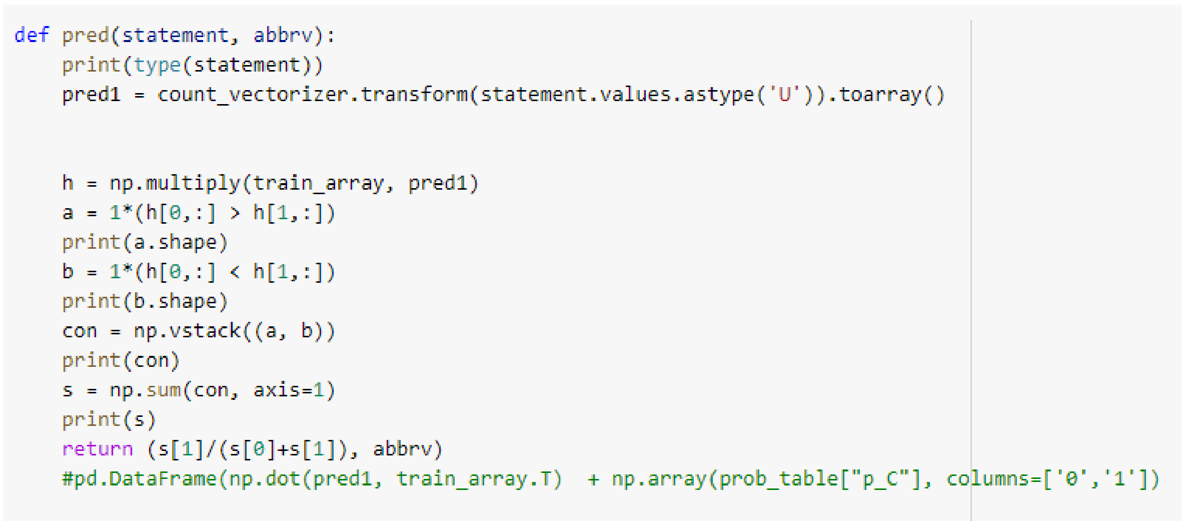
In our work size of the BOW Test is 2000 rows and 27,636 columns, the size of the training array is 2 rows and 27,635 columns, and the predicted data frame is 2000 rows and 2 columns. We have every sentence's original category and predicted category. So, we can see the results on the 2000 data points of BOW Test data we have got 1846 correctly classified data points. with 92.3% accuracy.

We have implemented our model now, and we have used it on our test data, in our test data we have created 178 rows with 4 columns which are an abbreviation, category, syntax, and sentence. First, we have dropped all the columns except sentences and store them as BOW Test1. we have used a count vectorizer.transform, so we use the same dimension as previous. 27635 no. of distinct words of the training data. so on the same dimension, we will use it for our text data.



```
predict_df_1 = pd.DataFrame(np.dot(BOW_test_1,train_array.T) + np.array(prob_table["p_C"]),columns=["0", "1"])
```

For our test data, we have to predict the category of the sentences. In the original test data, we have known the category of the sentences. We have given the input sentences and abbreviation, for that we have created a function and named this function as Score function, to decide for the given abbreviation, is COVID related or not by using the probability of a sentence being COVID related and non-COVID related.



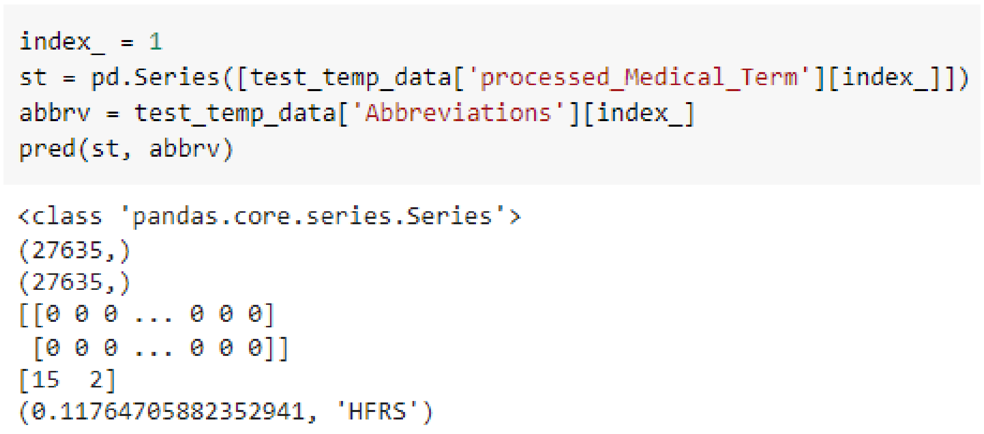
```
def pred(statement, abbrv):
    print(type(statement))
    pred1 = count_vectorizer.transform(statement.values.astype('U')).toarray()


    h = np.multiply(train_array, pred1)
    a = 1*(h[0,:] > h[1,:])
    print(a.shape)
    b = 1*(h[0,:] < h[1,:])
    print(b.shape)
    con = np.vstack((a, b))
    print(con)
    s = np.sum(con, axis=1)
    print(s)
    return (s[1]/(s[0]+s[1]), abbrv)
    #pd.DataFrame(np.dot(pred1, train_array.T)  + np.array(prob_table["p_C"], columns=['0','1'])
```

In the Score function, we have passed the argument statement and abbreviation. for the sentence, we have used the same dimension which is no. of distinct words of training data. Words of the given sentence matched with the words of corpus and created a vector if the word of sentence present in corpus assign it with '1' otherwise '0' and represented by pred. We have used 'h' to store the result of multiplication of train array and pred. we know train array is np array and contains the probability value for each word being COVID related and non-COVID related. So 'h' will represent a matrix with the size of the matrix based on the max size of train array and pred.

Now, we have got the probability of each word of the given sentence being COVID related and non-COVID related.
a = number of columns where the probability of non-COVID > COVID
b = number of columns where the probability of COVID > non-COVID
with the help of vstack function used vertically shows the value of a and b and is represented with the help of con. So, the probability of an abbreviation being COVID-related:
$=\frac{b}{a+b}$

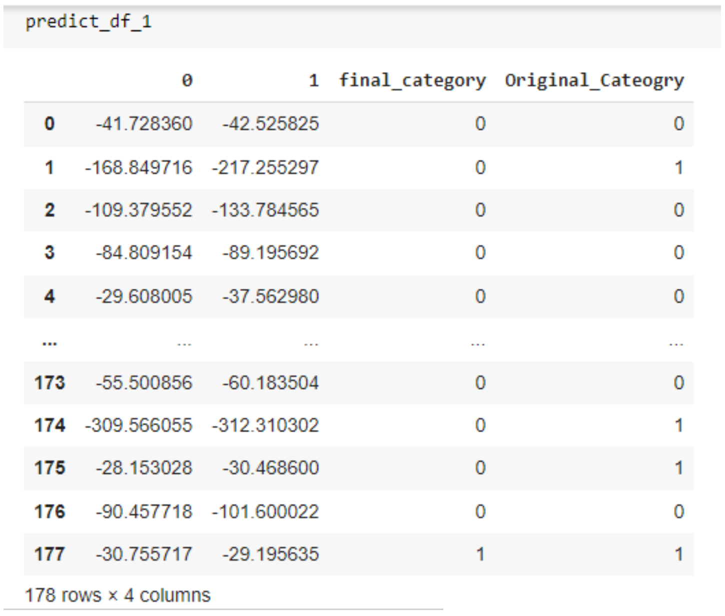
In our work, we got a = 15, and b = 2, so, the probability of the abbreviation being COVID related = 0.117, and the name of the abbreviation is HFRS.



```
index_ = 1
st = pd.Series([test_temp_data['processed_Medical_Term'][index_]])
abbrv = test_temp_data['Abbreviations'][index_]
pred(st, abbrv)
```

```
<class 'pandas.core.series.Series'>
(27635,)
(27635,)
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
[15  2]
(0.11764705882352941, 'HFRS')
```

from the above work, we will get to know whether the given abbreviation is COVID related or non-COVID related. if our abbreviation is COVID related then only we have assigned the COVID-related expansion to the corresponding abbreviation.



```
predict_df_1
```

	0	1	final_category	Original_Cateogry
0	-41.728360	-42.525825	0	0
1	-168.849716	-217.255297	0	1
2	-109.379552	-133.784565	0	0
3	-84.809154	-89.195692	0	0
4	-29.608005	-37.562980	0	0
...	...	...	...	...
173	-55.500856	-60.183504	0	0
174	-309.566055	-312.310302	0	1
175	-28.153028	-30.468600	0	1
176	-90.457718	-101.600022	0	0
177	-30.755717	-29.195635	1	1

178 rows × 4 columns

On the 178 data points of BOW Test data, we have received 105 correctly classified data points with 0.58% accuracy.

7 Conclusion

In this work, we developed methods for detecting whether an abbreviation of a given sentence is COVID-related or non-COVID-related. with the help of the probability of the sentence being COVID related and non-COVID related. So in the first work, we found sentence is COVID related or not and we got 95% accuracy on the train data. Hence, after using it on the test data which we have created, we got 57% accuracy with the multinomial Naive Bayes.

In the second work, we used the BOW approach and Laplace smoothing and we received 92.3% accuracy on train data and 58% accuracy on the independent test data. For our work, it is necessary to find the status of a given sentence is COVID-related or non-COVID-related.

8 Dataset and Code Availability

Dataset of COVID ontology: `https://github.com/COVID-19-ontology/COVID-19`

Dataset of general medical ontology: `https://conservancy.umn.edu/handle/11299/137703`

Codes from the GitHub link: `https://github.com/nidhipal076/Ontology-aware-learning-f`

References

[1] Muhammad Abbas, K Ali Memon, A Aleem Jamali, Saleemullah Memon, and Anees Ahmed. Multinomial naive bayes classification model for sentiment analysis. *IJCSNS Int. J. Comput. Sci. Netw. Secur*, 19(3):62, 2019.

[2] Olivier Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270, 2004.

[3] Biswanath Dutta and Michael DeBellis. Codo: an ontology for collection and analysis of covid-19 data. *arXiv preprint arXiv:2009.01210*, 2020.

[4] GW Geerling, M Labrador-Garcia, JGPW Clevers, AMJ Ragas, and AJM Smits. Classification of floodplain vegetation by data fusion of spectral (casi) and lidar data. *International Journal of Remote Sensing*, 28(19):4263–4284, 2007.

[5] Bob Grange and David A Bloom. Acronyms, abbreviations and initialisms. 2000.

[6] Trisha Greenhalgh. How to read a paper: the medline database. *Bmj*, 315(7101):180–183, 1997.

[7] Feng He and Xiaoqing Ding. Improving naive bayes text classifier using smoothing methods. In *European Conference on Information Retrieval*, pages 703–707. Springer, 2007.

[8] Sungrim Moon, Serguei Pakhomov, Nathan Liu, James O Ryan, and Genevieve B Melton. A sense inventory for clinical abbreviations and acronyms created using clinical notes and medical dictionary resources. *Journal of the American Medical Informatics Association*, 21(2):299–307, 2014.

[9] Marta Skreta, Aryan Arbabi, Jixuan Wang, Erik Drysdale, Jacob Kelly, Devin Singh, and Michael Brudno. Automatically disambiguating medical acronyms with ontology-aware deep learning. *Nature Communications*, 12(1):1–10, 2021.

[10] Mark Stevenson and Yikun Guo. Disambiguation of ambiguous biomedical terms using examples generated from the umls metathesaurus. *Journal of biomedical informatics*, 43(5):762–773, 2010.

[11] Peter Willett. The porter stemming algorithm: then and now. *Program*, 2006.

[12] Hua Xu, Peter D Stetson, and Carol Friedman. A study of abbreviations in clinical notes. In *AMIA annual symposium proceedings*, volume 2007, page 821. American Medical Informatics Association, 2007.

[13] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. Understanding bag-of-words model: a statistical framework. *International journal of machine learning and cybernetics*, 1(1):43–52, 2010.

[14] Wei Zhou, Vetle I Torvik, and Neil R Smalheiser. Adam: another database of abbreviations in medline. *Bioinformatics*, 22(22):2813–2818, 2006.