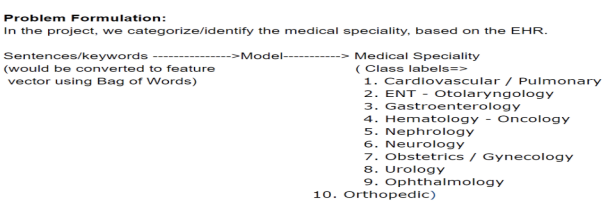
Medical Diagnosis using Patient's Notes

1214364306 1212787912 1213185088
Aditya Gupta and **Bhavani Balasubramanyam** and **Darsh Parikh** and
1214464576 1213083974
Kushal Reddy Papakannu and **Vaibhav Reddy kalakota**
Arizona State University
{agupt207, bbalasu6, dparikh2, kpapakan, vkalakot}@asu.edu

1 Introduction and Motivation

Health Care is one of the major areas where Natural Language Processing (NLP) is being widely used. Electronic Health Records (EHR) can improve the ability to diagnose diseases and prevent medical errors. With the increasingly broadening adoption of EHR worldwide, there is a growing need to widen the use of EHR data to support clinical research. Accurate diagnosis of EHRs to identify and suggest a cure for the disease without any involvement of a doctor would be a major breakthrough in the world of medicine. NLP and its techniques are used to classify diseases, symptoms, medicines and cure from the details obtained from EHR's.

In this paper, we use a combination of natural language processing techniques and machine learning techniques to develop models that can learn embedding of clinical terms and notes. We propose multiple models, and compare them on a wide range of evaluation tasks. We categorize/identify the medical specialty based on the above EHR information. At first, the data from MTSamples (`http://mtsamples.com/site/pages/`) is collected. Next, the data is parsed to extract all the possible words/phrases that may indicate a possible medical specialty symptom(s)/drug(s). Once the above information is available, different classification techniques are applied in order to classify the symptom/drug to a particular medical specialty.



2 Related Works

Health Care:
Ontology of medical concepts such as the Unified Medical Language System (UMLS) or the International Classification of Diseases (ICD-9, ICD-10) are widely used for epidemiology, health management, and clinical purposes, [1] proposed a method called Low Dimensional Representation(embedding) for medical diagnosis notes, [2] built two models which used all the available blood test parameters and the other used only a reduced set that is usually measured upon patient admittance to predict a hematologic disease using medical diagnosis notes, [3] in which they used SVM[4], Naive Bayes[5] to predict hypoglycemia among patients with [6] in which the authors used applied Random Forest (RF) and RRELIEFF [7] to evaluate a number of features, [8] used machine learning method that hybridizes support vector machine and simulated annealing.

Text Classification:
With the rapid growth of online information, text categorization has become one of the key techniques for handling and organizing text data. Text categorization techniques are used to classify news stories, and interesting information on the WWW, and to guide a user's search through hypertext. [9] analyzes the particular properties of learning with text data and identifies why SVMs are appropriate for this task, [10] describes the differences and details of multivariate Bernoulli model and multinomial model, [11] presents an empirical comparison of 12 feature selection methods, for text classification problem, [12] presents empirical results of Bayesian and decision tree machine learning algorithm on two text categorization datasets, [13] presents 2 feature

evaluation metrics for Naive Bayes algorithm, which is used to do text classification for multi-class text classification.

3 The Approach

Our aim is to develop a model, to predict the medical specialty, based on the EHR of the patient.

Data Preprocessing
Before implementing any of the approaches, we used the NLTK library in python[14], for removing stop-words, removing all non-alpha characters, converting text to lowercase and stemming the resultant data.Porter Stemmer[14] was used for stemming.

Approach 1:
In this approach we extracted data from MtSample datasets into an XML file containing 'Sample Type', 'Description', 'Sample Name' columns. Stopwords (the list of stopwords is taken from the one provided in nltk.tokenize) were removed from this file furthermore stemming(It is done using Porterstemmer, provided in nltk.stemmer) is performed to reduce the feature vector size and to decrease the computation time. A new column is made, which contains all the processed data.

Then we created a Bag of words[15] using:

1. Term Frequency (Tf)[16]
2. Term Frequency-Inverse Document Frequency (Tf-idf)[17]

Then, we provided the Bag-of-words representation of text to train Multi-class SVM and Naive Bayes algorithm both Tf and Tf-idf.

Approach 2:
In this approach we extracted four different columns namely 'Problem', 'Drug', 'Treatment' and 'Test' from MtSamples dataset. The program extracted individual columns from the XML file. Data from each column is combined into a list and then processed to remove stopwords and furthermore stemming is performed to reduce the feature vector size and to decrease the computation time. A new column is made, which contains all the processed data.

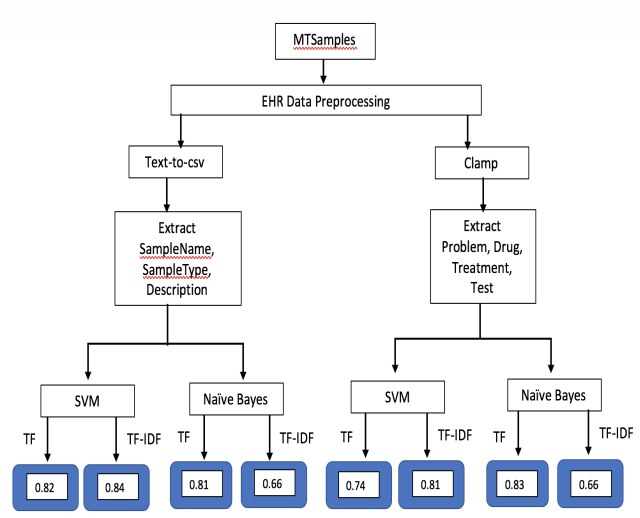
Then we created a Bag of words[15] using:

1. Term Frequency (Tf)[16]
2. Term Frequency-Inverse Document Frequency (Tf-idf)[17]

Then, we provided the Bag-of-words representation of text to train Multi-class SVM and Naive Bayes algorithm both Tf and Tf-idf.

Dimensionality Reduction
Initially, when we used Bag-of-words, to create features out of the text, we got around 12000 features when we used the whole document and 7000 features when we used the clamp data. To reduce this, we applied the following 2 approaches:

1. **PCA:** When implementing this, we first reduced the number of features to 80, which covered around 70% of the features. So, then we increased the number of features to 300, which, covered around 98% of the features
2. **Zero Variance Features:** We used this to remove the features that had zero variance. This helped us in filtering out the features that are actually useless for classification since they were present in all the documents.
3. **Chi-square:** To reduce the feature list further, we selected the best 300 features using the chi-squared statistics.



Dataset Creation
Our project title is "Medical Diagnosis using Patient's Notes". Patient notes are highly confidential, and hence initially it proved difficult to get the data we wanted. We looked at the following datasets:

1. `http://www.mtsamples.com/`

The above mentioned dataset, has patient notes classified according to the medical specialty.

Other datasets such as `https://www.kaggle.com/plarmuseau/sdsort/data` also has EHR data but it is difficult to obtain this due to IRB copyrights. Hence, we decided to use that data.

Data scraping

We built a web-scraper, using BeautifulSoup[18] and Selenium[19] web-framework to scrap the data of the following specialities:

1. Cardiovascular / Pulmonary
2. ENT - Otolaryngology
3. Gastroenterology
4. Hematology - Oncology
5. Nephrology
6. Neurology
7. Obstetrics / Gynecology
8. Urology
9. Ophthalmology
10. Orthopedic

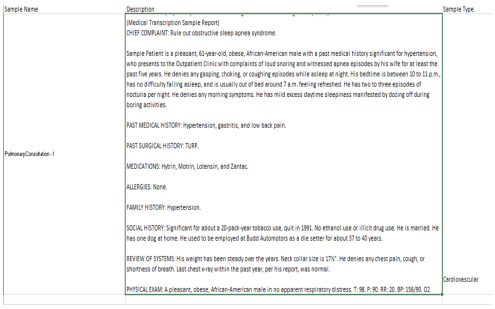
Selenium opened each page, having the patient notes, in a browser, and got the text of that page and saved it as a text file.

The dataset created has 1858 documents.

Data Files

To use machine learning algorithms, we needed to convert the data into proper format. So we created the following 2 dataset files:

1. The first dataset had the following columns:
   - Sample Name
   - Description (the Patient notes)
   - Sample Type



2. The second dataset had the following columns:
   - SampleType
   - SampleName
   - Problem
   - Drug
   - Treatment
   - Test

	A	B	C	D	E	F	G	H
1	SampleType	SampleName	problem	drug	treatment	test		
2	Urology	Cystourethrosc	right ureter	Coconut Oil	Ureteral stent-pla	Transurethral cystoscopy (procedure		
3	ENT - Otolaryngology	BMT - Bilateral	a small incision in the ea	Feeding tube, Pro	Temperature, Woman			
4	Cardiovascular / Pulmonary	Tracheostomy	dilation dist	anesthetic, (	Tracheostomy prc	Neck exploration, Flexible fiberoptic		

For creating the datafiles shown in the above figure, we used some of the tools developed by Mayo clinic for clinical analysis.

Initially we used MedTagger[20] and cTakes [21] medical parsers. These parsers are made by the Mayo Clinic research unit. Although useful to an extent, there were some software issues when we wanted to use the Side Effects or the Diagnosis annotators(in medTagger). Also, the documentation provided for medTagger is scarce. We then had come up with Clamp[22] [23], a software developed by UTH. We found that clamp has a unique feature where it can map relations between the entities. Example: Test Name: Heart Beat, Test Value: 72. Also, instead of considering only Unigrams, Clamp checks for Unigrams, Bigrams and N-grams looking for semantic meaning relating to medical terms.

Example output from Clamp

509 527 test C1821417 present Resting heart rate
531 533 labvalue null null 67
535 549 test C0005823 present blood pressure
553 559 labvalue null null 129/86

Example output from cTakes

<type:ConceptMention xmi:id="17633" sofa="6" begin="2004" end="2011" detectionMethod="DictionaryLookup" normTarget="heart" Certainty="Positive" semGroup="ANAT" status="Present" sentence="16495" experiencer="Patient"/>
<type:ConceptMention xmi:id="17644" sofa="6" begin="2030" end="2036" detectionMethod="DictionaryLookup" normTarget="rate" Certainty="Positive" semGroup="FIND" status="Present" sentence="16501" experiencer="Patient"/>
<type:ConceptMention xmi:id="17655" sofa="6"

begin="2049" end="2068" detectionMethod="DictionaryLookup" normTarget="blood" Certainty="Negated" semGroup="FIND" status="Present" sentence="16501" experiencer="Patient"/>
<type:ConceptMention xmi:id="17666" sofa="6" begin="2070" end="2075" detectionMethod="DictionaryLookup" normTarget="pressure" Certainty="Positive" semGroup="ANAT" status="Present" sentence="16513" experiencer="Patient"/>
<type:ConceptMention xmi:id="17677" sofa="6" begin="2070" end="2075"

Also, another advantage of using Clamp is the mapping between Lab Test and Lab Value. All the numerical values are ignored when we use cTakes or Medtagger. Clamp not only considers the numerical values but also maps them against their respective entities.

Final Sample output from Clamp
391 396 BDL null null Chest
418 427 drug C0001443, RxNorm=[296], Generic=[296] null Adenosine
433 445 test C0034606 present nuclear scan
479 490 test C0204014 present a treadmill
509 527 test C1821417 present Resting heart rate
531 533 labvalue null null 67
535 549 test C0005823 present blood pressure
553 559 labvalue null null 129/86
561 564 test C1623258 present EKG
592 600 drug C2343283, RxNorm=[795698], Generic=[795691] null Lexiscan
601 607 drug::STRENGTH null null 0.4 mg
609 619 test C0018810 present heart rate
624 626 labvalue null null 83
628 642 test C0005823 present blood pressure

XML Parser
To convert the xml files retrieved from ctakes/clamp and generate dataset files differentiating the different attributes found in respective columns, we needed a xml parser. To accomplish this, we used inbuilt python library - xml.etree.ElementTree - to build a xml tree. Then we iterated over the full tree to get the required data using semantic tags assigned by the medical parsers. We filtered the data in four columns - problem, treatment, drug, and test. For some of the features, we used the Unified Medical Language System (UMLS) description provided in the xml file as they give a more accurate explanation and more technical term of the medical text as opposed to what was provided in the file. This also helped us in maintaining some uniformity across the data.

Training Dataset
For training and validating the model, 80% documents from the dataset are taken as input.

Validation Dataset
For validating our model, we have used 10-folds cross-validation, using the Training data.

Test Dataset
For testing the model, we created a "test.csv" file, which contained 20% of the documents(from the dataset created).

4 Evaluation and Results

Dataset Used - Mtsamples

Test Training split
The model developed, would be trained with 80% of the documents from the dataset (20% of documents from dataset would form the test dataset). Furthermore, 10-fold cross-validation will be used, to train and validate the the model.

The accuracy will be measured by predicting class labels in 10-fold validation and dividing it with the total predictions made by the model.

Experimental Results
Our major aim to to classify the data based on the SampleName, Problem, Drug, Treatment and Test. When done using 10-fold validation, we were able to obtain around 75 to 85 percent accuracy which have gone till 88% when the split has been changed to 80-20 i.e. increase the amount of data to train the model.

Initial Hypothesis: Our initial hypothesis was, the keywords of the document (Problem, Drug, Treatment and Test), had higher significance, as compared to rest of the text in the document.

The results didn't really support our hypothesis. This could be seen by the results mentioned below:

Initially, to get the baseline accuracy, we used the document as a whole, created a vector from the features extracted using Bag of Words, and trained 2 models:

1. Naive Bayes - 0.8076774(TF)
2. Naive Bayes - 0.66024568 (TF-IDF)

3. SVM - 0.82391502(TF)
4. SVM - 0.84299228(TF-IDF)

Then, we parsed the data using Clamp. We extracted features like SampleType ,SampleName , Problem, Drug, Treatment and Test. We then created a vector, by combining all the features using Bag of Words and then trained 2 models:

1. Naive Bayes - 0.8252533 (TF)
2. Naive Bayes - 0.65938706(TF-IDF)
3. SVM - 0.73848676(TF)
4. SVM - 0.8089588(TF-IDF)

Other Inferences: Comparing SVM and Naive Bayes over both TF and TF-IDF models, SVM using TF-IDF is giving the highest accuracy over other combinations. Also, Naive Bayes under TF-IDF is the poorest performer over other models. This might be because SVM doesn't consider the different vectors to be independent of one another. Also, TF-IDF gives more weight to the terms that it deems as important. The combination of both - i.e. the feature matrix being dependent on each other and weights being allocated according to the term and inverse document frequency is giving us the best accuracy.

While taking metrics for each class we observed that classes like Neurology and Urology did not fare well when we compared with the results of other class. The reason can be due to the fact that there might be some similarities in the word(semantic relation) ex: Urology and Nephrology. Here Urology deals with urinary system and Nephrology deals with kidneys. There might be some common words or diseases that can be in both of these classes are used vice verse.

Result Tables

Class independent Results taking the document as a whole

	Accuracy	Precision	Recall	F1
Naïve Bayes - TF	0.8	0.81	0.77	0.78
Naïve Bayes - TF-IDF	0.68	0.65	0.46	0.51
SVM - TF	0.84	0.82	0.81	0.82
SVM - TF-IDF	0.85	0.82	0.83	0.86

Table1 *Class independent Results taking the document as a whole*

Class independent Results taking Clamp data

	Accuracy	Precision	Recall	F1
Naïve Bayes - TF	0.84	0.84	0.83	0.82
Naïve Bayes - TF-IDF	0.77	0.83	0.75	0.77
SVM - TF	0.81	0.79	0.76	0.75
SVM - TF-IDF	0.84	0.87	0.83	0.84

Table2 *Class independent Results taking Clamp data*

Class dependent Results taking the document as a whole

NB TF	precision	recall	f1-score
Cardiovascular / Pulmonary	0.84	0.81	0.84
Orthopedic	0.9	0.84	0.87
Gastroenterology	0.83	0.72	0.81
Neurology	0.52	0.63	0.61
Urology	0.67	0.75	0.73
Obstetrics / Gynecology	0.81	0.84	0.79
ENT - Otolaryngology	0.88	0.83	0.83
Hematology - Oncology	0.84	0.95	0.89
Ophthalmology	0.83	0.92	0.87
Nephrology	0.86	0.8	0.86

Table3 *Class dependent Results taking the document as a whole Naive Bayes Term Frequency*

NB TF-IDF	precision	recall	f1-score
Cardiovascular / Pulmonary	0.73	0.94	0.81
Orthopedic	0.91	0.75	0.81
Gastroenterology	0.87	0.82	0.9
Neurology	0.5	0.42	0.47
Urology	0.82	0.46	0.6
Obstetrics / Gynecology	0.8	0.77	0.76
ENT - Otolaryngology	0.87	0.88	0.87
Hematology - Oncology	1	0.82	0.9
Ophthalmology	0.81	0.92	0.89
Nephrology	0.89	0.84	0.86

Table4 *Class dependent Results taking the document as a whole Naive Bayes Term Frequency - Inverse Document Frequency*

SVM TF	precision	recall	f1-score
Cardiovascular / Pulmonary	0.91	0.97	0.94
Orthopedic	0.92	0.87	0.91
Gastroenterology	0.86	0.89	0.87
Neurology	0.7	0.62	0.71
Urology	0.8	0.86	0.81
Obstetrics / Gynecology	0.98	0.96	0.95
ENT - Otolaryngology	0.98	0.89	0.93
Hematology - Oncology	0.96	0.91	0.95
Ophthalmology	0.92	0.93	0.92
Nephrology	0.94	0.87	0.91

Table5 *Class dependent Results taking the document as a whole SVM Term Frequency*

NB TF-IDF	precision	recall	f1-score
Cardiovascular / Pulmonary	0.77	0.92	0.84
Orthopedic	0.97	0.73	0.83
Gastroenterology	0.8	0.8	0.8
Neurology	0.63	0.52	0.57
Urology	0.86	0.47	0.61
Obstetrics / Gynecology	0.84	0.72	0.78
ENT - Otolaryngology	0.89	0.84	0.86
Hematology - Oncology	0.98	0.77	0.86
Ophthalmology	0.79	0.96	0.87
Nephrology	0.9	0.83	0.87

Table8 *Class dependent Results for Clamp Naive Bayes Term Frequency - Inverse Document Frequency*

SVM TF-IDF	precision	recall	f1-score
Cardiovascular / Pulmonary	0.79	0.97	0.88
Orthopedic	0.98	0.78	0.84
Gastroenterology	0.77	0.84	0.82
Neurology	0.72	0.68	0.74
Urology	0.82	0.48	0.42
Obstetrics / Gynecology	0.99	0.71	0.81
ENT - Otolaryngology	0.94	0.95	0.92
Hematology - Oncology	0.96	0.83	0. 91
Ophthalmology	0.86	0.98	0.92
Nephrology	0.83	0.8	0.81

Table6 *Class dependent Results taking the document as a whole SVM Term Frequency - Inverse Document Frequency*

SVM TF	precision	recall	f1-score
Cardiovascular / Pulmonary	0.89	0.95	0.92
Orthopedic	0.96	0.85	0.9
Gastroenterology	0.86	0.89	0.87
Neurology	0.87	0.83	0.85
Urology	0.86	0.82	0.84
Obstetrics / Gynecology	0.96	0.94	0.95
ENT - Otolaryngology	0.96	0.89	0.92
Hematology - Oncology	0.97	0.92	0.94
Ophthalmology	0.93	0.98	0.95
Nephrology	0.95	0.89	0.92
avg / total	0.92	0.92	0.92

Table9 *Class dependent Results for Clamp SVM Term Frequency*

Class dependent Results taking Clamp data

NB TF	precision	recall	f1-score
Cardiovascular / Pulmonary	0.88	0.83	0.85
Orthopedic	0.92	0.81	0.86
Gastroenterology	0.87	0.74	0.8
Neurology	0.59	0.73	0.65
Urology	0.66	0.77	0.71
Obstetrics / Gynecology	0.84	0.81	0.82
ENT - Otolaryngology	0.86	0.82	0.84
Hematology - Oncology	0.83	0.92	0.87
Ophthalmology	0.84	0.93	0.88
Nephrology	0.85	0.83	0.84
avg / total	0.84	0.83	0.83

Table7 *Class dependent Results for Clamp Naive Bayes Term Frequency*

SVM TF-IDF	precision	recall	f1-score
Cardiovascular / Pulmonary	0.75	0.96	0.84
Orthopedic	0.98	0.59	0.74
Gastroenterology	0.7	0.92	0.8
Neurology	0.95	0.48	0.64
Urology	1	0.18	0.31
Obstetrics / Gynecology	0.99	0.61	0.76
ENT - Otolaryngology	0.92	0.91	0.91
Hematology - Oncology	0.96	0.82	0.89
Ophthalmology	0.85	0.99	0.92
Nephrology	0.82	0.87	0.84

Table10 *Class dependent Results for Clamp SVM Term Frequency - Inverse Document Frequency*

5 Challenges Faced

While preparing the dataset, we faced some issues. Those are:

1. Initially, we built the web-scraper only using BeautiFulSoup, which went to each page, having the patient notes, and saved the page, and then later parsed the data from the saved HTML files to make the corresponding text files. But, due to varied response time from the website, many pages (almost 50%) did not get saved and we were not able to get all the data. To solve the above problem, we then decided to use selenium, along with BeautifulSoup.

2. While parsing the XML files created by Clamp,we faced the issue of there being no tag for the category and the name of the disease, and they were simply listed as medical terms like all others. So we had to do a regexp search on the full document to retrieve those.

3. While preprocessing the data, we applied PCA, for dimension reduction, and got some negative values. The negative values caused issue, when we inputted those values in Naive Bayes. To counter this, for Naive Bayes, if the value was < 0, that value was replaced by 0 and if the value was > 0, it was replaced by 1.

Error/Challenges Examples

1. BOW output [0 0 0 0 1 0 0 1 0 0 . . . . 0 0 0] converts to [-1.27 -2.2 -1.7 -0.8 0.26 -0.8 0.2 0.3 -1.78 0.2 . . . . 0.162 -0.002 1.06]
   The final vector cannot be used as an input feature for our Naive Bayes model as it takes only non negative values

2. As shown in the previous example, Sentence like "Heart rate has increased" will change into 'Heart','rate','has','increased' instead of splitting it into 'Heart rate' ,'increased'

6 Conclusion

In this work, initially, we built a dataset of EHRs, classified according to medical specialty that they belonged. We then trained 2 machine learning models (SVM and Naive Bayes),based on 2 approaches, to predict the medical speciality, given the EHR record of the patient. The 2 approaches are:

1. Training the models of the entire EHR
   - Tf
   - Tf-idf
2. Training the models only on the keywords
   - Tf
   - Tf-idf

Our initial hypothesis was, that the keywords of the document(Problem, Drug, Treatment and Test), had higher significance, as compared to rest of the text in the document, in predicting the medical specialty of the EHR of patient.

Based on the accuracy, we found that our initial hypothesis was wrong, as we got a higher accuracy for the first approach, as compared to the second approach.This might be due to the following reasons:

1. The medical speciality, (of a given the EHR), may have higher dependency on text other than the keywords

2. The tool we have used, Clamp to extract the keywords, is not 100% accurate and thus it did not give perfect data, which might have affected the accuracy.

7 Future Work

Though, we got a good accuracy, using basic machine learning algorithms [SVM and Naive Bayes], more work could be done in this area. In future, we might explore tools similar to Clamp, which give better results, in terms of medical term extraction.

For representing the text data, we have used Bag-of-words model. In future, we could use other vector space models like word2vec, doc2vec etc. can be used to learn the word embedding, which might result in better classification of the documents.

We have only used basic machine learning models to do the classification. We could implement some deep-learning models like CNN, RNN to do the classification.

Lastly, in this work, we have just predicted the medical specialty, based on EHR. Furthermore, our model is based on data which is very specific. In future, we could enhance the model, so that it is able to work with different types of data, and we could also work on training the model to predict the disease(based on the EHR), instead of the the medical specialty.

References

[1] Choi, Y., Chiu, C. Y. I., Sontag, D. (2016). Learning low-dimensional representations of medical concepts. AMIA Summits on Translational Science Proceedings, 2016, 41.

[2] Guncar, G., Kukar, M., Notar, M., Brvar, M., Cernelc, P., Notar, M., Notar, M. (2018). An application of machine learning to haematological diagnosis. Scientific reports, 8(1), 411.

[3] Cryer, P. E., Davis, S. N., Shamoon, H. (2003). Hypoglycemia in diabetes. Diabetes care, 26(6), 1902-1912.

[4] Cortes, C., Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

[5] http://scikit-learn.org/stable/modules/naive_bayes.html

[6] Ferraty, F., Vieu, P. (2003). Curves discrimination: a nonparametric functional approach. Computational Statistics Data Analysis, 44(1-2), 161-173.

[7] Robnik-Sikonja, M., Kononenko, I. (2003). Theoretical and empirical analysis of ReliefF and RReliefF. Machine learning, 53(1-2), 23-69.

[8] Nazari, I., Moghaddam, F. A., Zamani, M. M., Salimi, J. (2012). Clinical characteristics and remedies in 45 Iranians with carotid body tumors. Acta Medica Iranica, 50(5), 339.

[9] Joachims, T. (1998, April). Text categorization with support vector machines: Learning with many relevant features. In European conference on machine learning (pp. 137-142). Springer, Berlin, Heidelberg.

[10] Thatcher, N., Chang, A., Parikh, P., Pereira, J. R., Ciuleanu, T., von Pawel, J., ... Carroll, K. (2005). Gefitinib plus best supportive care in previously treated patients with refractory advanced non-small-cell lung cancer: results from a randomised, placebo-controlled, multicentre study (Iressa Survival Evaluation in Lung Cancer). The Lancet, 366(9496), 1527-1537.

[11] Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. Journal of machine learning research, 3(Mar), 1289-1305.

[12] Lewis, D. A., Campbell, M. J., Terry, R. D., Morrison, J. H. (1987). Laminar and regional distributions of neurofibrillary tangles and neuritic plaques in Alzheimer's disease: a quantitative study of visual and auditory cortices. Journal of Neuroscience, 7(6), 1799-1808.

[13] Yavuz, M. S., Cheng, Y., Chen, J., Cobley, C. M., Zhang, Q., Rycenga, M., ... Wang, L. V. (2009). Gold nanocages covered by smart polymers for controlled release with near-infrared light. Nature materials, 8(12), 935.

[14] https://www.nltk.org/

[15] https://deeplearning4j.org/bagofwords-tf-idf

[16] https://nlp.stanford.edu/IR-book/html/htmledition/term-frequency-and-weighting-1.html

[17] https://nlp.stanford.edu/IR-book/html/htmledition/inverse-document-frequency-1.html

[18] https://www.crummy.com/software/BeautifulSoup/

[19] https://www.seleniumhq.org/

[20] http://ohnlp.org/index.php/MedTagger

[21] http://ctakes.apache.org/

[22] http://clamp.uth.edu/

[23] Ohno-Machado, L., Sansone, S. A., Alter, G., Fore, I., Grethe, J., Xu, H., ... Soysal, E. (2017). Finding useful data across multiple biomedical data repositories using DataMed. Nature genetics, 49(6), 816.