# Predicting Gene-Disease Association Type in Biotechnology Literature using Text Mining

**R.M.U.A. Rathnayake**

**2019**

# Predicting Gene-Disease Association Type in Biotechnology Literature using Text Mining

A dissertation submitted for the Degree of Master of Science in Computer Science

R.M.U.A. Rathnayake
University of Colombo School of Computing
2019

# Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: R.M.U.A. Rathnayake

Registration Number: 2016/MCS/094

Index Number: 16440947

Signature: Date: 25-10-2019

This is to certify that this thesis is based on the work of

Mr. R.M.U.A. Rathnayake

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Dr. A. R. Weerasinghe

Signature: Date:

# Abstract

Extracting Gene-Disease relations using text mining techniques and machine learning is a popular and important task in the present research context. As a single person can not read a bunch of papers to derive conclusions about a certain topic, text mining and machine learning models help a lot in this case. Revealing gene-disease associations in brute force approach is more accurate but involves in lengthy laborious work, effort intensive and resource intensive. Associations between gene and disease entities mentioned within biomedical literature can be used to develop new drugs development, diagnosis mechanisms and treatment mechanisms in western medicine.

This study focuses on research questions: How to extract gene-disease associations from biomedical literature, How to build a predictive model for gene-disease association using machine learning and How to validate the predictive model. The aim is to predict gene disease association type (positive or negative) mentioned within the sentences. Genetic Association Database (GAD) and PubMed abstracts were used in training models and validating them. Training and validating the models carried out based on three cycles. In all three cycles, Naive Bayes (NB), Linear support vector machine (LSVM) and a logistic regression (LR) classifier models were trained and evaluated.

In the first cycle of the training process, model training and validation tasks were based on entirely GAD. 70% of this data set used for training the models. In the second cycle PubMed abstracts were used to create a new testing data set.1000 sentences with gene and disease mentions were manually labeled according to the relationship type. Next, the models trained during the first cycle were used to predict the entire new data set. In the third cycle the models were re-trained entirely on GAD (100%) and again investigate the model accuracy.

We obtained good results in predicting gene-disease association types in biomedical literature. The maximum accuracy achieved by NB, LSVM and LR classifiers were, 0.876, 0.951 and 0.929 respectively. Predicted results then visualize through a network graph which shows the predicted relationships between genes and diseases. Moreover, the querying interface was provided to query the output or the resulted findings where by providing either gene entity or disease entity of interest. This will allow any interested party to see multiple associations predicted for a particular entity.

As a whole, the aim of the study is to derive a predictive model to identify gene-disease relations in human related biomedical literature using text mining and machine learning.

# Acknowledgements

This thesis would have been a dream for me without support extended by the following noble people as it is because of them I completed this thesis with great confidence and passion.

I would like to convey my sincere gratitude to the supervisor of the research study, Dr. A. R. Weerasinghe, a senior lecturer and former director of University of Colombo School of Computing (UCSC), Mrs. Rupika Wijesinghe, the core-supervisor, and all the staff members of the UCSC for supporting and encouraging me to become a mindful academic.

I also like to convey my special thanks to Dr. Lasanthi De Silva, who was the coordinator of masters research projects for the immense support given throughout the period of research study.

I would like to extend my thanks to all other masters colleagues who willingly shared their beliefs, past experiences and insights about the research work I engaged in.

It is my great pleasure to acknowledge all the evaluators and examiners who share their views and valuable insights on my research from the very beginning.

My family members who were always wishing my well-being should be appraised and rewarded with many thanks for baring all the difficulties with me.

# Table of Contents

# List of Tables

# List of Figures

FIGURE Page

# Abbreviations

**LSVM** Linear Support Vector Machine

**LR** Logistic Regression

**NB** Naive Bayes

**NER** Named Entity Recognition

**HGNC** Hugo Gene Nomenclature Committee

**NIEHS** National Institute of Environmental Health Sciences

**NLP** Natural Language Processing

**NCBI** National Center for Biotechnology Information

**CNN** Convolutional Neural Network

**GAD** Genetic Association Database

**CTD** Comparative Toxicogenomics Database

# Chapter 1

# Introduction

At present, it is crucial to identify the linkage between genes and diseases in advance so that the medical practitioners and researchers can concentrate on identifying mechanisms to prevent, diagnose and most importantly to come up with treatments for diseases associated with genes. It is worthwhile to address this issue of revealing possible gene-disease associations based on the published literature.

Exponential growth of published literature in the form of electronic documents, leads to difficulties in properly storing, managing and extracting useful and most up-to-date information [1]. In addition to that, the challenge of developing tools and methods for transforming heterogeneous data coming from various sources to structured biomedical knowledge is also important [1]. Hence uncovering hidden information out of extensive collections of literature has become more challenging than ever in early days [2].

It is impossible for a human to refer and stay up-to-date about all the literature of interest due to the alarming rate of growth in literature [3]. That is why computational techniques and methods are essential to assist the process of systematic retrieving, extracting and discovering relevant information from extensive collections of literature available. Statistics have shown that 500,000 papers are newly added every year to the MEDLINE database, a collection of abstracts in biomedical research. In 2010 this rate has been increased up to two papers per minute.

## 1.1 Background to the research

With the advancement of biomedical research, the focal point of attention has been changed from individual gene and protein related research to entire genome or biological systems based researches [3]. As a result of this shift, the importance of uncovering hidden relationships in huge collections of unstructured text has grabbed the attention of state-of-the-art researches. For example, it is crucial to identify the association between genes and diseases in advance so that the medical practitioners and researchers can concentrate on identifying possibilities to prevent, diagnose and most importantly to come up with treatments for diseases associated with genes. Discovering the possible hidden relationships among biological entities (e.g. proteins, genes, etc.) in advance prior to time, cost and effort intensive laborious work, not only simplifies the process of establishing new knowledge but also saves a lot of resources. As the rapid development in biomedical domain paved the way to exponential growth in number of publications, the life for a researcher has become so difficult and almost impossible to sync with all updates under a particular sub-field of interest for uncovering hidden associations among biomedical entities. Therefore, finding ways of automating the relation extraction process has become more popular in order to address this problem. Automating the screening process of unstructured text, transforming unstructured text to structured text and relation extraction help reducing time and effort significantly compared to manual approach. As a result many computational techniques have proposed for shedding a light to overcome the aforementioned problem.

Among the computing techniques proposed, an array of different approaches with different success rates can be observed in literature. These vary from simple co-occurrence based to more advanced machine learning and deep learning based approaches at present.

## 1.2 Scope of the study

Scope of this research focuses on machine learning and text mining domain to create a better predictive model of gene-disease associations in freely available bio-medical literature abstracts. The attention is given towards the sentence level relations. When extracting the relationships, we consider a sentence at a time. Further, human genes and diseases were selected for the study. Further, relationships that are extended beyond a single sentence will not be considered under this study.

## 1.3 Research problem

Traditional text mining based models for gene-disease relations extraction heavily depend on word dictionaries and rules defined by the researchers. In other words such systems explicitly define keywords to identify gene entities and disease entities. This leads to the difficulty of maintaining and keeping such keywords up to date. As this updating process require time and effort it does not happen frequently. Further, the new keywords or entity names in recent publications would not be identified by such systems as they are not present in the dictionary. Hence, such systems failed to identify most recent gene-disease associations published on literature.

On the other hand, Convolutional Neural Network (CNN) based unsupervised learning models require a large data set with higher accuracy to obtain better predictions. And such systems depend on high end computing resources.

## 1.4 Aim of the study

The aim of the research study is to output predictive model to identify gene-disease relations in human related biomedical literature using text mining and machine learning.

## 1.5 Research questions

- How to extract gene-disease associations from biomedical literature?
- How to build a predictive model for gene-disease association using machine learning?
- How to validate the predictive model?

# Chapter 2

# Literature Review

Mining literature has become a trend in recent past among researchers who are fascinated by discovering interesting findings based on freely available literature sources. Manipulating what is already available to derive unseen but possible relationships among entities or concepts contribute to the expansion of interesting and challenging dimensions to think different when doing research.

According to [4], relation extraction (RE) approaches depend on Natural Language Processing (NLP) techniques and Machine Learning (ML) techniques. Moreover, the nature of the techniques used in relation extraction in general, it can be further sub-divided in to co-occurrence, pattern, rule and machine learning based approaches.

NLP driven approaches consist of syntax and semantic analysis phases which helps solving most of the ambiguities related to text processing. Mainly NLP does transform unstructured textual data in to a structured form. Further any relation extraction system can be studied under three main sub sections namely preprocessing module, parser module and RE module. NLP comes in preprocessing stage which prepares appropriate input form of text to the parser module [4]. Preprocessing stage involves in sentence splitting, tokenizing, Part-of-Speech (POS) tagging and Named Entity Recognition (NER) [4]. See Figure 2.1.

Sentence splitting separates individual sentences from an input text (e.g. abstracts, paragraphs, etc.) typically using appropriate delimiters (e.g. periods, question marks, exclamation marks, etc.). [4] elaborates that the nature of biomedical publication conventions (e.g. irregularity in biomedical entity names (e.g. E.coli), abbreviations (e.g. et al.) and in-line citations (e.g. Sci. 2006) breaks the straight forwardness of the sentence splitting step and makes it a challenging task. Due to this reason, these sentence splitters need to be re-trained on biomedical corpora such as GENIA for a better accuracy prior using in biomedical domain [3]. Tokenization step is used to split the sentences in to tokens generally based on white

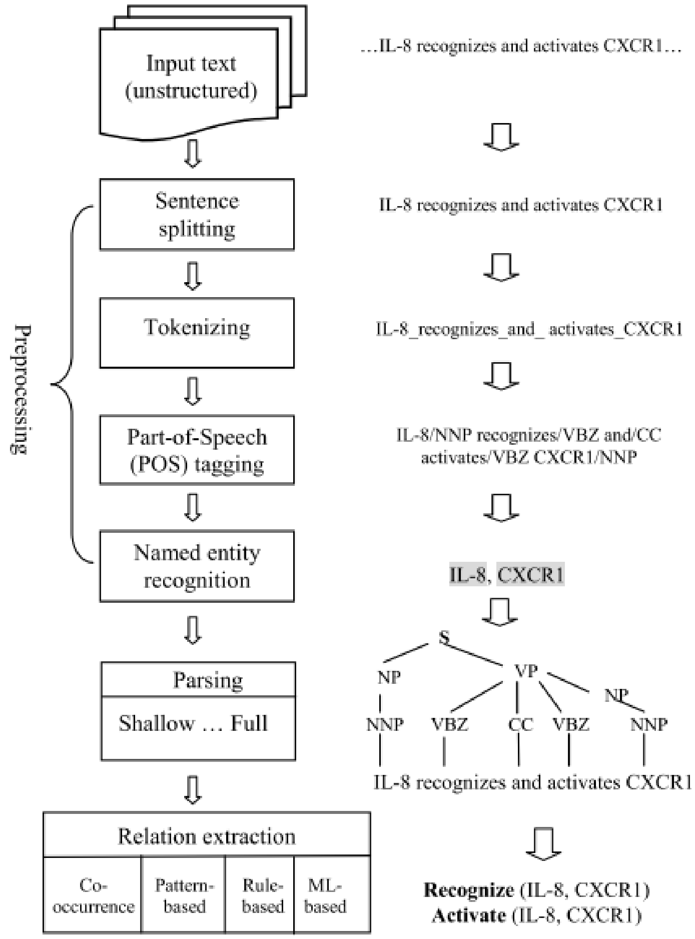

Input text (unstructured)
Sentence splitting
Tokenizing
Part-of-Speech (POS) tagging
Named entity recognition
Parsing
Shallow … Full
Relation extraction
Co-occurrence
Pattern-based
Rule-based
ML-based
Preprocessing
…IL-8 recognizes and activates CXCR1…
IL-8 recognizes and activates CXCR1
IL-8_recognizes_and_ activates_CXCR1
IL-8/NNP recognizes/VBZ and/CC activates/VBZ CXCR1/NNP
IL-8, CXCR1
S
NP
VP
NP
NNP
VBZ
CC
VBZ
NNP
IL-8 recognizes and activates CXCR1
Recognize (IL-8, CXCR1)
Activate (IL-8, CXCR1)

FIGURE 2.1. Work flow of a typical RE system, [4]

space characters. Tokenization step is also subjected to disturbances caused by domain specific terminology, non-standard punctuations and orthographic patterns [5]. In addition to that, hyphenation introduces ambiguities in determining the number of tokens to be returned by the tokenizer depending on the context. It is important to note that the errors in early stages of NLP process affect the accuracy of parser and RE modules of any RE system [4].

Each lexical item or a word can be assigned under a category defined by the language (e.g. verb, noun, adjective, etc.) [5]. This refers as POS tagging and it assigns the grammatical form of the word after identifying the context through the surrounding words and word itself [4].

NER task often divided in to two sub tasks: first, recognition of words that refer to the entities and second, the unique identification of the entities [3]. Identifying biomedical entities is a prerequisite for any relation extraction system which deals with biomedical domain. Though NER looks trivial, it is one of the most difficult and accuracy sensitive tasks as all the successor steps of a RE system rely on it [6]. Lack of standardization of naming conventions of biomedical entities has set the sense for listing NER as one of the daunting tasks. Having several different names for the same biomedical entity, the same name can refer to different entities depending on the context and having multi-word names for certain biomedical entities are the issues arising from lack of standardization of names [4]. Manconi identifies the need of standardization as one of the factors that require significant attention to enhance future research in biomedical domain [2]. Garten summarizes three main approaches for NER, lexicon based, rule based and statistical or ML based respectively [6].

- lexicon based approach: entity identification is based on lexicons or dictionary entries present in text [6].
  - main drawback of word dictionary based approach is the inability to capture new names introduced in biomedical literature (word-order variations in newly introduced entities matter). This cleared the way to encourage rule based approaches [6].
- rule based approach: uses patterns or rules to match with text content in literature [6].
  - This approach depends on manually furnished rules and patterns which makes it difficult to adapt for a different context or a sub domain of interest in a particular domain (e.g. sub-domains of biomedical domain also have a significantly difference in naming entities). Because of this, rule based approach is also given less attention opening the avenues for ML based approaches [6].
- statistical or ML based approach: expects annotated corpora (considered to be golden standards) as input. ML techniques are capable of automatically identifying entities in text to determine positive set and negative set based on features previously learned through training [6].
  - [6] mentions, Hidden Markov Models, Support Vector Machines (SVM) and Conditional Random Fields (CRFs) as commonly used types of machine learning methods in this regard.

According to [4] parsing step is sub divided in to two abstract categories namely, shallow parsing and full parsing. Instead of resolving the structure of the sentence to the level of single elements, shallow parsing works with annotated chunks of words or phrases. Further, these systems are powered by rule based to ML based approaches [4]. The full parsing output of a given sentence is a parse tree which reveals the relationship of subject-predicate-object structure of the sentence and three types of full parser systems can be identified [4].

- phrase structure parsing: A tree structure of a sentence, having syntactic tags as nodes and words in the leaves.
- dependency parsing: A tree structure of a sentence, of which words represent nodes and edges represent relationships among words.
- deep parsing: Represents syntactic and semantic structures and predicate argument structures.

Machine learning techniques help finding solutions to the problems in bioinformatics domain [1]. In the paper, typical problems associated with bioinformatics have been classified in to six different problem domains [1]. Those domains represent genomics, proteomics, microarrays, systems biology, evolution and text mining. The paper extends on how supervised classification, clustering, probabilistic graphical models and optimization techniques can be used to satisfy the pre-identified problem domains. It is mentioned that the text mining techniques are more important in the areas of Cellular location prediction, functional annotation and uncovering protein interactions to extract knowledge from existing literature.

Text mining involves in detection, extraction, and maintenance of knowledge. These steps based on Information Retrieval (IR), Information Extraction (IE) and Entity Recognition (ER) in general. IR is about finding papers related to a given query. Still, the one who is searching for literature will have to run through a series of documents listed to shortlist the relevant ones which impose an extra burden. ER is all about finding the biological entities mentioned in the text. The biological entity may be a protein, gene, etc. IE process deals with the relationships among entities identified [3].

## 2.1 Deep learning interventions for relation extraction

An interesting research [7] uses CNN driven method for extracting relations from literature. A typical CNN structure in relation extraction is shown in figure 2.2. However this study emphasizes that previous studies of others heavily depend on features generated by manual linguistic modules or supervised NLP toolkits. As a drawback, paper further explains that these erroneous features produced by the third party tools contribute to the presence of errors in RE and relation classification (RC). In order to minimize the dependency with external toolkits, authors have introduced unsupervised framework to learn features automatically through screening sentences in literature.

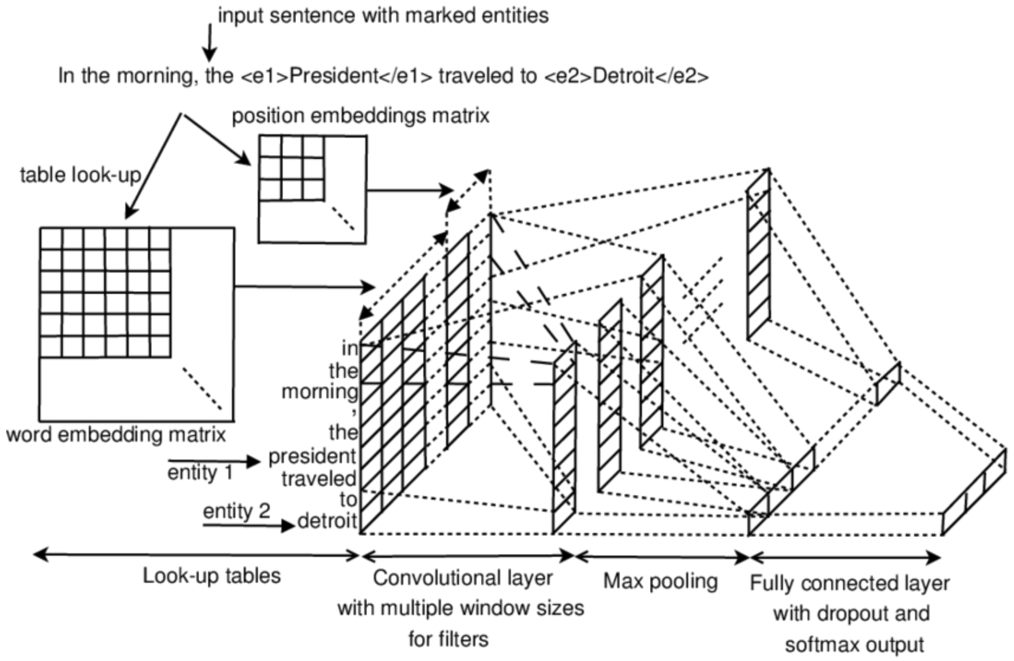

input sentence with marked entities
In the morning, the <e1>President</e1> traveled to <e2>Detroit</e2>
position embeddings matrix
table look-up
word embedding matrix
entity 1
entity 2
in
the
morning
,
the
president
traveled
to
detroit
Look-up tables
Convolutional layer with multiple window sizes for filters
Max pooling
Fully connected layer with dropout and softmax output

FIGURE 2.2. Convolutional Neural Network for Relation Extraction, [7]

Concretely, position embedding matrix and a word embedding matrix derived from input sentences with marked entities are concatenated to form the word vectors. As the next step, filters with multiple window sizes have been applied in convolutional layer for a better coverage of n-grams. Max pooling strategy is applied for each filter to generate a set of abstract pooling scores from which then forms the feature vector. Moreover, a dropout vector is produced from the feature vector to fetch in to the fully connected layer and at last a soft max layer performs final classification task. In addition to both RE and RC experiments of the study are successful, [7] emphasizes that RE is more challenging than RC after investigating the performance variation with an unbalanced data set (ACE 2005 data set). In conclusion, the proposed CNN method outperforms all the baseline work in literature. This research study points out following future work remains for further research aspects [7].

- enrich the representation of CNNs with more features for RE
- study applications of CNNs in other related tasks
- examining other neural network models for RE

Lee and the research team during the study on “Deep learning of mutation-gene-drug relations from the literature” presents two new computational methods based on the PubMed articles for the curation of mutation-gene-drug relations [8]. The first method uses a machine learning classifier. Biomedical Search Entity

Tool (BEST) scores have been used as some of the features to train classifier. In the second method uses a combination of BEST scores and word vectors to train a Convolutional Neural Network (CNN) model. The mutation-gene and mutation-drug relations are extracted based on the random forest classifier and CNN. The model of this study learns by itself and then operate based on the derived knowledge. As the system automatically learns about the relations, this model seems to be more appealing to an environment having updated with new articles frequently. Therefore, the dynamic nature of the resource base to be mined to extract knowledge would not be a primary barrier.

Similarly, another research proposes a solution for automated detection of adverse drug reactions (ADR) by screening literature having the aim of accurately distinguished between the ADR relevant and irrelevant documents [9]. A CNN and biomedical word embeddings based approach has been used to achieve good performance over the traditional and Long Short Term Memory (LSTM) models. This study is based on a previous research [10]. In both researches, authors have incorporated Adverse Drug Effects (ADE) corpus which is a benchmark corpus developed by Gurulingappa to support automatic extraction of drug related adverse effects [11]. De-duplication of ADR relevant sentences in ADE dataset, improving performance by integrating word embedding specifically developed for biomedical text and comparing two existing CNN architectures are listed as key contributions of the research work.

## 2.2 Cosine similarity between vectors for relation prediction

Moreover another recent study has been carried out to predict the associations between genes and diseases. This study evaluates the cosine similarity between gene vectors and disease vectors to uncover possible linkages. Vectors are constructed based on the appearance and the location of the gene disease terms mentioned in the abstract articles of PubMed database. Term weight (TW) and co-occurrence methods along with MeSH database and TF-IDF methods have been integrated for seeking better efficiency and accuracy in predictions. Introducing weight matrix, penalty for keyword (PWK) and normalization aspects are considered in evaluating the performance of the new method. In the prediction performance evaluation, the method proposed in this paper outperforms Heterogeneous Network Edge Prediction (HNEP) method with higher precision and recall. It is also mentioned that the results of the study can be integrated with other models for gaining improved performance in gene-disease predictions [12].

## 2.3 Recent development for relation extraction

Biomedical Entity Search Tool (BEST) which is an advanced tool for directly returning target entities from literature rather than a long list of articles in contrast to standard tools like PubMed. The study explains that it is difficult to manually process the increasing amount of published literature by humans, text mining techniques and tools have been developed to assist users. Many of these tools which fall in to the category of biomedical search entity tools are developed to enhance PubMed search. Further, outdated results, slow response time and limited coverage were three main limitations observed in such tools [13] . Paper highlights the ability of BEST to process free text queries and return up-to-date results in real time.

HiPub [14] is a chrome browser plug-in capable of automatically identifying, annotating and translating biomedical entities from text and forming networks for knowledge discovery purposes. It is mentioned that biomedical entities like proteins, genes, diseases, drugs, mutations and cell lines can be identified with high precision and recall. This tool utilizes two main named-entity recognition tools: PubTator and BEST entity extractor.

BeFree is another gene-disease relations extraction system based on supervised learning. In this study the authors have used morphological and syntactic features of text. Further they have used a dependency kernal for explaining the gene-disease relations extraction application in translational research. This tool consists of a Biomedical Named Entity Recognition (BioNER) that is based on dictionaries with FUzzy and pattern matching methods. Further EU-ADR and GAD corpora have been used in this study.

DISEASES is another resource for extracting gene disease associations from biomedical literature. This tool uses a dictionary based tagger in the process of identifying named entities (genes and diseases) in humans. Further it looks at the co-occurrences within sentences and between sentences of the literature. This paper provides reasons for using dictionary based approach against machine learning method as the necessity of having a high quality gene and disease names dictionary in which names can be normalized to reduce the number of false positives [15].

But when considering the nature of the entity names of interest the suitability of the technique used for NER purposes may vary. In overall different techniques can be identified in to three main categories each having distinct advantages.

- Rule-based: names with a strongly defined orthographic and morphological structure;

- Dictionary-based: closely defined vocabulary of names (e.g. diseases and species);

- ML-based: strong variability and highly dynamic vocabulary of names (e.g. genes and proteins).

According to the specific applications, dictionary based approach is better for identifying diseases while ML based approach is better for identifying gene names. In DISEASES it commonly used dictionary based approach for NER purpose. And also these rule-based and dictionary based approaches fail to detect new terminologies with a fair accuracy against its simplicity for implementation and the straightforward nature [16].

Know-GENE is another effort to uncover gene-disease relationships by combining co-occurrence based gene-gene mutual information integration with known protein-protein interactions. Boosted tree regression method has been used for the prediction of associations [17].

Moreover, PKDE4J, a comprehensive text mining system. This employed rule based relations extraction technique and for the purpose of NER, it uses dictionary based approach and for relations extraction it uses rule-based approach [16]. Again in this study only focus on single method of entity extraction which is not well suited for both entity types (genes and diseases). Dictionary based method is better for disease names. This might reduce the accuracy of detecting gene names which will influence the final relations extraction performance of the system.

## 2.4 Summary

The proposed methodologies for gene disease relation extraction in above, lacked a well-crafted supervised machine learning approach based on gold standard corpora. Among the works discussed above most of them followed a dictionary based tagging and a rule-based relation extraction. Only the two systems BeFree and DTMiner used a machine learning approach for relation extraction and reported results on EU-ADR and GAD corpora.

Depending on the entity type, the approach for better identification of a particular type of entity would be different. The approaches like rules based, dictionary based and machine learning based could be used to train models for capturing named entities. Each approach has different level of capabilities and suitability for the targeted entities. When looking at gene and disease entity recognition, gene names come with high variability. This makes a gene vocabulary highly dynamic in most of the cases. In order to address the challenge of such nature ML based approach for training the model would be ideal rather than a dictionary based approach. Further, the disease names are closely defined in general and such a vocabulary can be trained using a dictionary based approach. However, most of the recent research in the text mining in bio-medical literature have used the power of ML techniques for the purpose of training models for NER and relations extraction.

In addition to that, deep learning techniques such as Convolutional Neural Networks based studies have proven to be to have promising results in the text mining domain. The only barrier is the requirement of larger collections of corpus for properly training such models. One of the main compelling issues in the biomedical domain is that lack of standardization and huge variations in naming conventions of biomedical entities. And also due to the complex nature of the associations among the biomedical entities, it is much more difficult to complete the annotation process rapidly. This led the existing large scale corpora suffer from quality issues.

However, it is needed to explore possibilities of extending machine learning based gene-disease association extraction in order to improve the extraction and curation of genetic association of diseases.

# Chapter 3

# Research Methodology

## 3.1 Introduction

This chapter explains the research methodology followed throughout the study. Research methodology explained using two main diagrammatic representations under figure 3.1 and figure 3.2. Figure 3.1 illustrates the model training process using an existing data set (GAD) acquired through literature. 3.2 illustrates the steps followed in creating the new manually curated data set based on PubMed abstracts and how the trained models were validated. In this study, we trained three machine learning models, Naive Bayes (NB), Linear Support Vector Machine (LSVM) and Logistic Regression (LR). Evaluation conducted in three cycles. First, the same GAD data set was used for both training and validating the models. 70% of the data were used in training and 30% percent of the data used in testing or validating the models. Secondly, The trained model used to validate a manually curated data set based on PubMed articles. In the third cycle, all the models re-trained entirely based on GAD data set (100%) and those new models were evaluated base on the prediction accuracy on the newly created PubMed data set.

## 3.2 Training and validating the models based on Genetic Association Database (GAD) data set

Figure 3.1 presents the flow of steps in training machine learning classifiers based on the GAD data set. All the three, NB, LSVM and LR models used for training and validating purposes.

### 3.2.1 Nature of the GAD data set

GAD data set consists of two files GAD_Y_N.csv and GAD_F.csv. We used the data available in GAD_Y_N.csv file. There were 2802 records of interest. Each record is

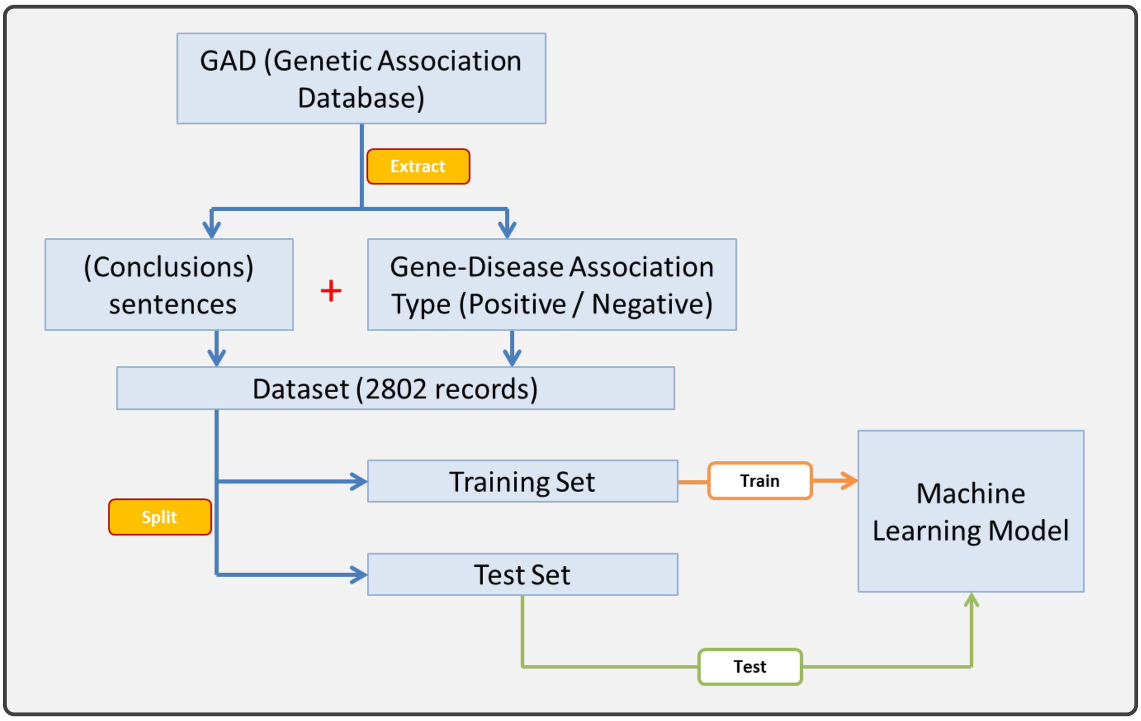


FIGURE 3.1. Research methodology diagram 1 (Train classifier models using GAD data set)

defined using eleven columns. All the column names and the meanings are listed below.

- GAD_ID: Identification record from GAD
- GAD_ASSOC: Type of association (Y, N or F)
- GAD_GENE_SYMBOL: Gene symbol provided by GAD record
- GAD_GENE_NAME: Gene name provided by GAD record
- GAD_ENTREZ_ID: Entrez GeneID provided by GAD record
- NER_GENE_ENTITY: Gene text in the sentence provided by BioNER
- NER_GENE_OFFSET: Gene text offset in the sentence provided by BioNER at 'sentence level'
- GAD_DISEASE_NAME: Disease name provided by GAD record
- NER_DISEASE_ENTITY: Disease text in the sentence provided by BioNER
- NER_DISEASE_OFFSET: Disease text offset in the sentence provided by BioNER at 'sentence level'
- GAD_CONCLUSION: Sentence provided by GAD record

### 3.2.2 Training machine learning models with GAD data set

In this study, as shown in the figure 3.1, column number 2 and 11 were extracted to prepare the initial data set to train the model. Sentences were cleaned to reduce the processing overhead and to improve the accuracy. All bad symbols which are not interested have been removed using regular expressions and converted in to lower case. Stop word removal is performed on the data to further shrink down the data volume. In this step the original stop words list from nltk corpus was modified to avoid removing words that play a key role in generating negative sense of the sentence (eg: u"don't", u'has', u"haven't", u'not', u'nor', u"wasn't",u'didn', u"isn't",u"hasn't",u"doesn't",u"aren't", u"no)

Data set was split in to two subsets from which one set is used to model the train and the other set is to validate the trained model's accuracy to determine the suitability of using it for predicting association type (positively related or negatively related) in PubMed abstract sentences. Data set was split based on 30% to 70% percent ratio where 30 percent is for validation and 70 percent is used in training the model.

Naïve Bayer's classifier has been trained and as a base line model. After that, LSVM, and LR models were trained to assess the viability of the model.

## 3.3 Steps followed to create the new data set based on PubMed abstracts

The motivation of preparing a new data set is to properly evaluate the trained models. Figure 3.2 illustrates the steps followed to create the new data set. As the models were trained based on an existing data set and evaluated based on the same data set, we extracted a set of 1000 sentences from 1,893,815 (one million eight hundred ninety-three thousand eight hundred and fourteen) sentences. Due to the huge number of sentences, split the file in to sub files to which each sub file contains 200,000 sentences. Preparing a new data set using PubMed abstracts consists of following main sub steps.

- downloading abstract texts from PubMed and preparing sentences list.
- creating a list of all human gene names (HGNC symbols)
- creating a list of human disease names (a cancer related disease name list)
- filter sentences list against gene name list and disease name list to extract sentences where at least a single gene and disease name exist.
- annotate/label the sentences based on the relationship type

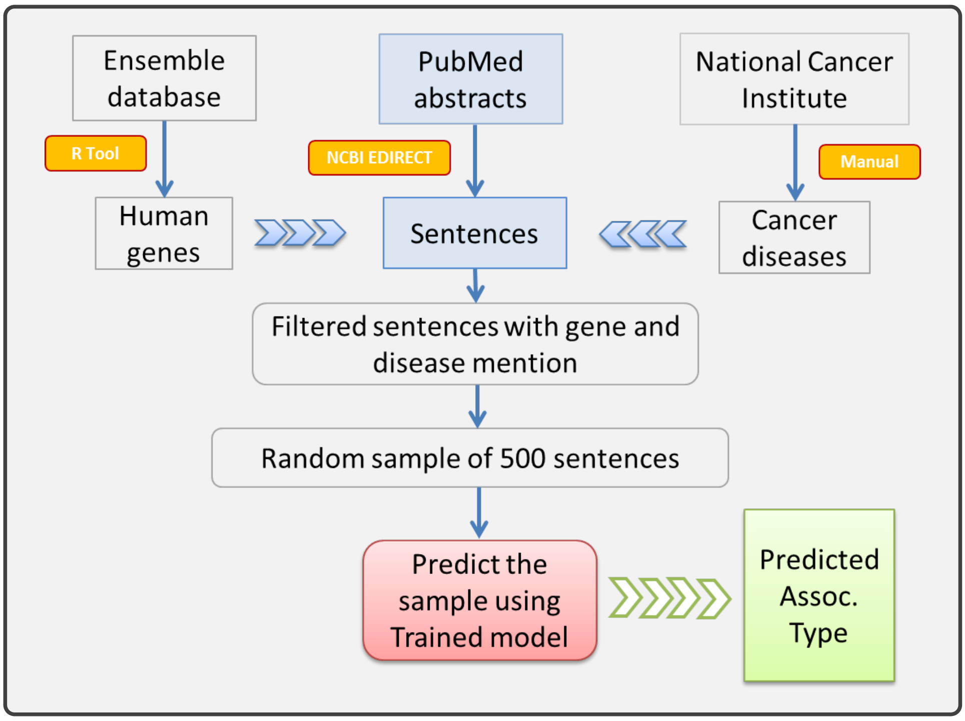

Ensemble database
R Tool
Human genes
PubMed abstracts
NCBI EDIRECT
Sentences
National Cancer Institute
Manual
Cancer diseases
Filtered sentences with gene and disease mention
Random sample of 500 sentences
Predict the sample using Trained model
Predicted Assoc. Type

FIGURE 3.2. Research methodology diagram 2 (Prepare new data set and evaluate using models)

Each step in preparing the new data set is explained in detail using the subsections below.

## 3.4 Downloading abstract texts from PubMed and preparing sentences list

In order to prepare a data set for the purpose of text mining to uncover gene-disease relations type, a source of literature documents needs to be selected. For this study the focus is on PubMed abstracts. PubMed has earned a large popularity among the research community and frequently updated the content which is freely available. As the abstract section of the published papers contain a rich summary of the documents content, this would also be an ideal section for identifying the key concepts discussed in the body of the paper. In a way it is challenging as the abstracts contain an overall summary of the work. Within the overall summary it may not provide information about details of associations between entities. In order to gather a good data set(sentences with gene-disease mentions) a large pool of abstracts needed. One of the main steps is to collect sufficient number of sentences from the PubMed abstracts that have valid gene and disease mentions.

National Center for Biotechnology Information (NCBI) is the party which maintains over 18 databases related to biotechnology domain. Other than providing

Graphical Web User Interface for accessing the data stored on these databases, it also provides a rich Application Programming Interface (API) for Mac, Unix and Linux based environments to access the same content programmatically. E-Utils API is the one used to access and filter out the data in the format required. This has been simplified further by introducing EDIRECT tool which is capable of communicating effectively with the E-utils API hiding the complexity to the programmer.

Using this EDIRECT tool, 204106 abstracts downloaded. "gene" AND "disease" query was used to filter the records using EDIRECT. These sample abstracts were split into sentences and total number of sentences were 1,893,815. These sentences were again split in to multiple files each having 200,000 records/sentences. The first two files containing 200000 sentences were used in preparing sample data set for this research.

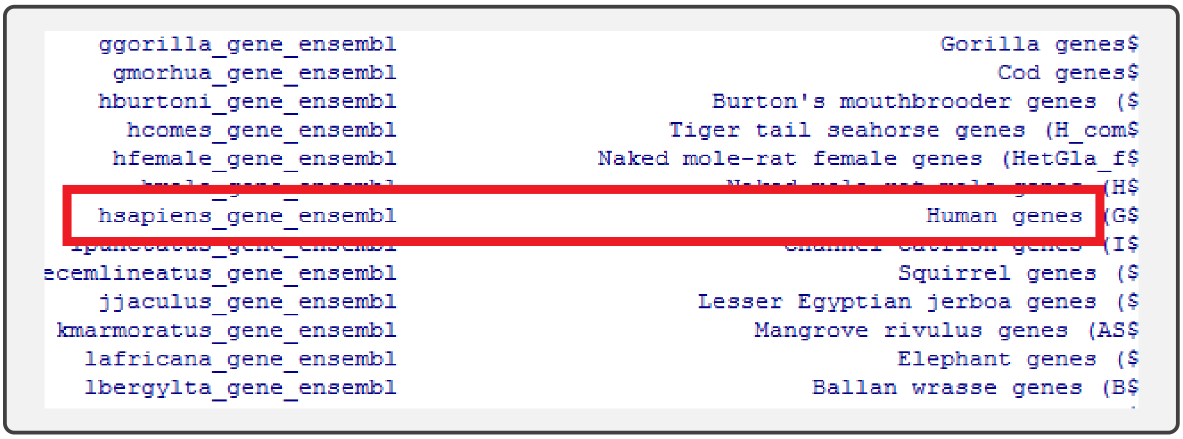
### 3.4.1 Creating a list of all human gene names (HGNC symbols)

Preparing a genes list that contains all human gene symbols is the focus under this section. This is prepared for the purpose of selecting out the sentences from the sentence pool that contain elements from this gene list. if a particular sentence contains elements from this list, we can ensure that a gene mention is within the particular sentence.

Ensemble genome browser is one of the major on-line tools for querying gene information. It contains different data sets including human gene data set. A data set refers to a particular species. Genes are organized in to chromosomes. As humans have 23 chromosome pairs, all genes collections can be downloaded through 24 files as there is a different X and Y for the last pair. After downloading genes related to all 24 chromosomes, concatenated the genes in to a single file to prepare a single list of all genes. R tool has been used to download the genes data set.

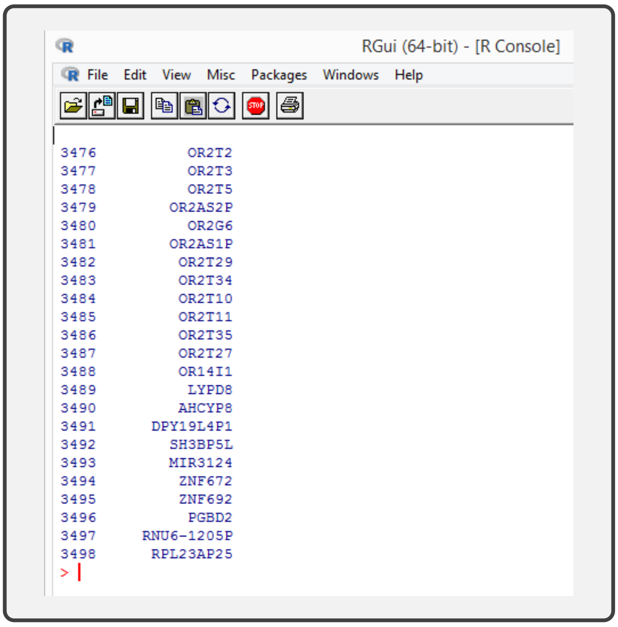
As there are multiple data sets available in Ensemble, figuring out the data set for humans was needed. See Figure 3.3. Further, figure 3.4 shows the output after completing the last command to get all the genes listed in human Chromosome 1. In overall it contained 3498 genes. This approach has been followed to remaining 23 chromosomes and downloaded the genes (HGNC) symbol list in comma separated (CSV) files.

Downloaded 24 files that contain HGNC symbols of all the genes in humans were further processed to form a single file containing all the gene symbols. All gene symbols were loaded in to a list. Pandas, a popular Python package for data



```
      ggorilla_gene_ensembl                              Gorilla genes$
       gmorhua_gene_ensembl                                  Cod genes$
      hburtoni_gene_ensembl                Burton's mouthbrooder genes ($
        hcomes_gene_ensembl                 Tiger tail seahorse genes (H_com$
       hfemale_gene_ensembl            Naked mole-rat female genes (HetGla_f$
         hmale_gene_ensembl              Naked mole-rat male genes (H$
      hsapiens_gene_ensembl                              Human genes (G$
    ipunctatus_gene_ensembl                         Channel catfish genes (I$
ecemlineatus_gene_ensembl                             Squirrel genes ($
     jjaculus_gene_ensembl                 Lesser Egyptian jerboa genes ($
 kmarmoratus_gene_ensembl                      Mangrove rivulus genes (AS$
     lafricana_gene_ensembl                           Elephant genes ($
     lbergylta_gene_ensembl                       Ballan wrasse genes (B$
```

FIGURE 3.3. Ensemble human genes data set definition


RGui (64-bit) - [R Console]
File Edit View Misc Packages Windows Help

```
3476         OR2T2
3477         OR2T3
3478         OR2T5
3479       OR2AS2P
3480         OR2G6
3481       OR2AS1P
3482        OR2T29
3483        OR2T34
3484        OR2T10
3485        OR2T11
3486        OR2T35
3487        OR2T27
3488        OR14I1
3489         LYPD8
3490        AHCYP8
3491     DPY19L4P1
3492       SH3BP5L
3493       MIR3124
3494        ZNF672
3495        ZNF692
3496         PGBD2
3497    RNU6-1205P
3498      RPL23AP25
> 
```

FIGURE 3.4. Genes in human chromosome 1

science, was used in the overall list preparation task. Processing steps are listed under appendix A.

### 3.4.2 Creating a list of human disease names (a cancer related disease name list)

Preparing the disease names vocabulary was more challenging due to lack of consistency in naming conventions among different diseases. Looking at several data sets from different sources that contain disease mentions consist of common words like "and", "child", "development", etc. Few example disease name mentions from Comparative Toxicogenomics Database (CTD) [18], maintained by MDI Biological Laboratory  NC State University with the funding support of National Institute of

Environmental Health Sciences (NIEHS) is listed to emphasize the diverse naming conventions.

Epstein-Barr Virus Infections, Chemical and Drug Induced Liver Injury, Child Development Disorders, Pervasive and Intervertebral Disc Degeneration are combinations of multiple words. Chemical and Drug Induced Liver Injury contains the words Chemical, and, Drug, Induced, Liver and Injury words can not consider individually as diseases. But, as a whole illustrates the disease condition. On the other hand Erythema, Cholangiocarcinoma, and Asthma gives the full meaning and the definition of the disease condition. Such diseases are quite obvious to capture among the words in a sentence at once. Preparing a concise disease vocabulary had to work on different data sets extracting the disease mentions. Comparative Taxogenomic Database (CTD) also contained a huge set of disease mentions.

The scope of the diseases narrowed down to cancer disease types other than focusing on entire diseases. Depending on the success of the results, this can be expanded into other disease types in future. However, cancer related diseases names were extracted from National Cancer Institute listing of cancer disease types. The unprocessed disease names were stored in in cancer_diseases.txt file. Processing original names involved in removing any punctuation marks among disease names which were defined using multiple words and other names provided using punctuation marks. For example, *"Fibrous Histiocytoma of Bone, Malignant, and Osteosarcoma"* is defined in a single line. Punctuation marks removed using *translate()* function.

```
disease.translate(string.maketrans("",""), string.punctuation)
```

On the other hand, line read from the file would contain multiple disease definitions as in the previous example. Hence, *nltk.word_tokenize()* function was used to prepare tokens. Prepared tokens wrote in to a csv file (diseases.csv). Next, all the tokens loaded into a pandas data frame and dropped duplicate tokens using *drop_duplicates()* function. Further, the tokens converted in to lower case and sorted for convenience in later stages of use in the research. Finally, the modified data frame saved into cancer.csv file. A list of disease names then created using the contents of the cancer.csv.file.

### 3.4.3 filter sentences list against gene name list and disease name list to extract sentences

First two set of PubMed abstract sentences (first two files with 200,000 sentences each) were filtered against both gene and disease lists prepared in section 3.4.2 and 3.4.1. The resulting sentences consist of 12844 sentences from the first 200,000 sentences and 14716 from the next 200,000 sentences. Hence we filter the sentences based on the gene and disease lists, the resulted sentences would contain at least one valid gene and disease mention. 500 sample sentences from each output were randomly selected so that to extract a set of 1000 sentences. Reservoir sampling algorithm was used to automate the sampling process using jupyter notebook in Python language. Relevant python programs can be found in appendix A section.

### 3.4.4 Annotate/label the sentences based on the relationship type

Extracted sample sentences were manually read to identify the relationship type. If the mentioned gene/s and the disease/s have a positive relationship, the sentence has labeled as "Y". For example in the sentence *"Somatic mutations in the isocitrate dehydrogenase 2 gene (IDH2) contribute to the pathogenesis of acute myeloid leukaemia (AML) through the production of the oncometabolite 2-hydroxyglutarate (2HG)1-8."* is considered as "Y". The sentence clearly expresses the positive sense of the relationship between IDH2 gene and acute myeloid leukaemia.

If the relationship is negative, labeled the sentence as "N". For example, the sentence *"Our findings did not suggest any association between CYP2A6 genotypes and risk of lung cancer."* provides the sense that there is no relationship between CYP2A6 and lung cancer. Hence such sentences were labeled as "N". During this labeling process certain sentences found where there is neither positive nor negative sentence. For example, the sentence *"CCL2 implication has not been investigated in canine urothelial carcinoma."* does not provide a good clue about the relationship. Such sentences were labeled as "F". For this study, "F" type has not been included. In this manner all five hundred sentences were labeled to prepare training data set for the purpose of classifying the relationships of unknown sentences.

## 3.5 Extending the model of the study to predict the degree of relationships (Strong positive or weak positive)

Accurate predictions of the models further annotated with the labels describing the level of positiveness of the relationship. The true positive sentences were labeled as "S" if the relationship is strong. For example, the sentence "Many studies have demonstrated that the genetic variants of tumor suppressor gene TP53 contribute to the prediction of breast cancer risk" demonstrates a strong relationship. The part of the sentence "Many studies have demonstrated" adds weight to the positive nature of the relationship mentioned in the sentence. For the weak case of relationship, the sentence was labeled as "W". For example, in the sentence "Therefore, IRF2 may be a potential target for AML treatment", the words "may be suggests the uncertainty of the relationship among gene and disease entities mentioned.

When annotating the degree of positiveness of the relationships, we annotated with extra detail like date of publication of the paper, PubMed identification number of the publication.

## 3.6 Evaluating the models and presenting the results

The prediction accuracy of the models were evaluated in three different cycles of evaluation. First, the models were trained using GAD data set and evaluated on the same data set. Secondly, the models in the previous cycle were evaluated against the prediction accuracy with newly created data set which is based on PubMed. Finally, the models were re-trained entirely on GAD data set and tested the resulting model prediction accuracy against the PubMed data set.

The results were presented to the user in two different forms. Developed a querying interface for the users to find the associations between gene and disease entities. The query supports either entities. If the passing query is a gene entity, it will shortlist all the diseases positively associated with that gene and the corresponding paper identification number in PubMed, date of publication and the degree of the relationship mentioning whether it is a strong positive or a weak positive. If the query entity is a disease, it will show the same details for the corresponding gene entities.

In order to view the results of multidimensional associations or relationships among gene and disease entities, the results were presented as a graph representation. Gene and disease entities considered as nodes in the graph and edges represent the relationships. Using Pandas python library, prediction results were read in to a data frame. Networkx library in python helps in creating a list of edges and nodes from the data frame object and Matplotlib visualization library was used to draw the network graph.

# Chapter 4

# Results and Evaluation

The chapter highlights the results derived throughout the study and interpretation of the results to emphasize on validity of the predictive models. Results are presented based on three cycles explained in section 3.1. The evaluation of the results will also discuss at the end of the chapter.

## 4.1 Results

This section provides results for three classification models. They are NB, LSVM, and LR. Resulted classification reports are described for each model in each study cycle under separate sub sections 4.1.1, 4.1.3, and 4.1.4

### 4.1.1 Cycle 1: Prediction results of the models on GAD data set

Initially the all three models were trained using the GAD data set which consists of 2802 records. The data set was divided in to a training set and a testing set which have proportions 70% and 30% of the total data set respectively. The data set distribution is shown based on the classes "Y" and "N" in 4.1.

As an initial baseline model the Naïve Bayes (NB) classifier was trained and predicted the unseen test data set. The prediction accuracy score for the test set was 0.81 as shown in table 4.1.

In the training set there were 841 records out of which 552 records correspond to presence of relationship between gene and disease mentioned in the sentence (classified as 'Y') and 289 sentences correspond to not related (classified as 'N').

Linear support vector machine (LSVM) training results are shown in table 4.2. LSVM has a good accuracy compared to the baseline model. Compared to the precision scores in table 4.1, the precision scores in table 4.2 are better in both 'Y' and 'N' cases.

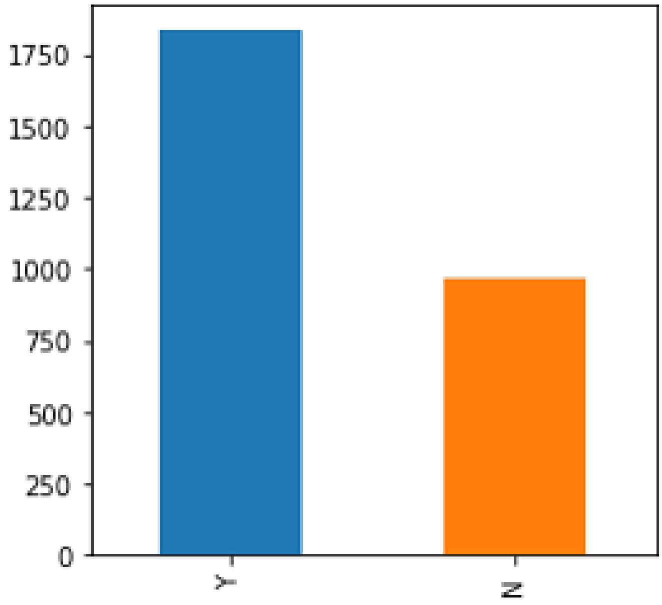


FIGURE 4.1. Data set distribution based on the class labels

Table 4.1: NB classifier prediction accuracy results - Cycle 1

**Accuracy 0.8145065398335315**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| N | 0.93 | 0.50 | 0.65 | 289 |
| Y | 0.79 | 0.98 | 0.87 | 552 |
| micro avg | 0.81 | 0.81 | 0.81 | 841 |
| macro avg | 0.86 | 0.74 | 0.76 | 841 |
| weighted avg | 0.84 | 0.81 | 0.80 | 841 |

Table 4.3 is related to the Logistic Regression (LR) classifier related accuracy score. It also ended up giving 91% overall accuracy which is similar to that of LSVM model. Just looking at the precision and recall, LR model has a slight improvement over SVM model. For the 'N' scenario both models confirm to equal figures (precision 0.90 and recall 0.84). Considering 'Y' class, precision score remains same at 0.92 and recall has a improved in LR model. There was no notable difference in f-score for LSVM and LR model predictions over test data set without advanced cross validation. However, both LSVM and LR models demonstrate a significant improvement compared to the Naive Bayes model.

Table 4.2: LSVM classifier prediction accuracy results - Cycle 1

**Accuracy 0.9143876337693222**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| N | 0.90 | 0.84 | 0.87 | 289 |
| Y | 0.92 | 0.95 | 0.94 | 552 |
| micro avg | 0.91 | 0.91 | 0.91 | 841 |
| macro avg | 0.91 | 0.90 | 0.90 | 841 |
| weighted avg | 0.91 | 0.91 | 0.91 | 841 |

Table 4.3: Logistic regression classifier prediction accuracy results - Cycle 1

**Accuracy 0.9167657550535078**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| N | 0.91 | 0.84 | 0.87 | 289 |
| Y | 0.92 | 0.96 | 0.94 | 552 |
| micro avg | 0.92 | 0.92 | 0.92 | 841 |
| macro avg | 0.92 | 0.90 | 0.91 | 841 |
| weighted avg | 0.92 | 0.92 | 0.92 | 841 |

### 4.1.2 10-Fold cross validation results

Table 4.4: Mean accuracy after 10-fold cross validation on classifier models

| Classifier | Cross validation scores | Mean score |
|---|---|---|
| **NB** | [0.83629893 0.80714286 0.82142857 0.81785714 0.77142857 0.79285714 0.83214286 0.81071429 0.82857143 0.8 ] | 0.8118441789527198 |
| **LSVM** | [0.91459075 0.91428571 0.91785714 0.93214286 0.90714286 0.91428571 0.92142857 0.94642857 0.93928571 0.93214286] | 0.9239590747330961 |
| **LR** | [0.91814947 0.90714286 0.92857143 0.93571429 0.93214286 0.91428571 0.925 0.96428571 0.95357143 0.93928571] | 0.9318149466192172 |

After the initial training cycles, as the next step all three classifiers were trained based on K-Fold cross validation. Set the number of folds into 10 in this case and set up a 10-fold cross validation where in each iteration the data set splits in to 10 folds. Ninety percent (90%) of the data is used for training and 10% is used to validate or test the model. This process continues until all the ten folds were validated against the rest of the data set. Table 4.4 show the mean accuracy of each classifier after the 10-fold cross validation based training step.

Results of the 10-fold cross validation step has improved in LSVM and LR models except NB model. LSVM has improved the accuracy from 0.91 (see table 4.2) to 0.92 and LR model has improved from 0.91 (see table 4.3) to 0.93.

### 4.1.3 Cycle 2: Prediction accuracy results on PubMed data set

Under cycle 2, the same models used in cycle 1 were used for the prediction. The entire data set based on PubMed has been considered as the test set. The data set distribution is shown in figure 4.2. Out of the 1000 records labeled, 727 records were considered excluding the third class labeled as "F".

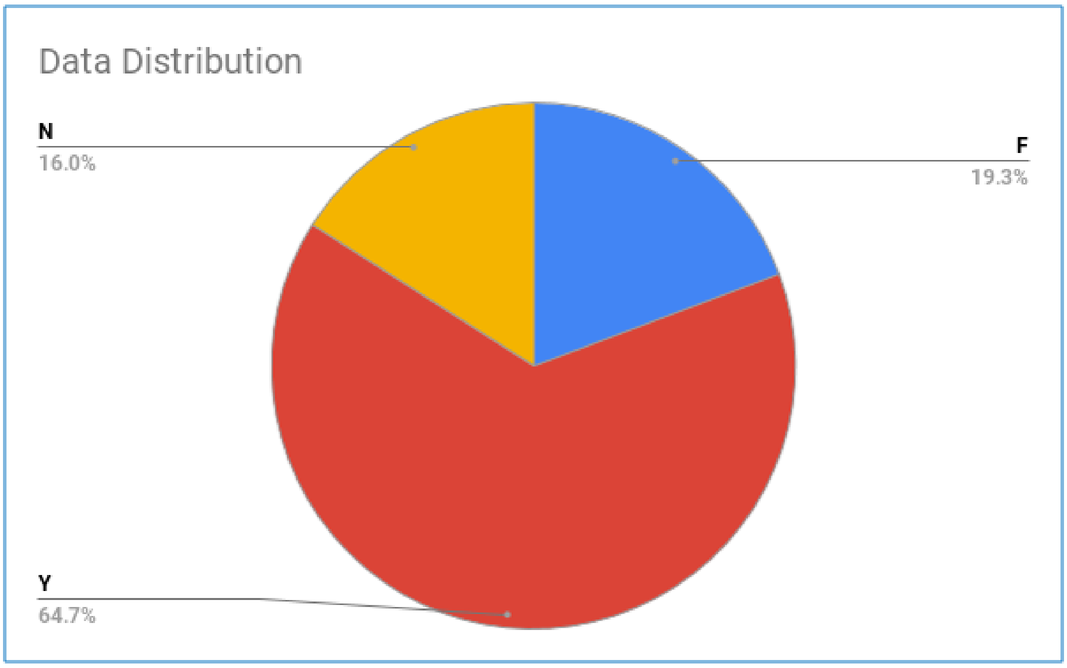


FIGURE 4.2. PubMed based data set distribution based on the class labels

During the labeling process we found that there are three classes available. Sentences with a positive relationship, sentences with a negative relationship and a neutral category. For example, the sentence "We show that PHF8 is **upregulated and positively correlated** with MYC at protein levels in breast cancer" was considered as a positively related one which was labeled as "Y" and the sentence "These results suggest that **there is no overall association** between rare alleles of the HRAS1 VNTR and breast cancer" was considered as negative ones labeled with "N". The sentences like "We investigated whether functional polymorphisms of CYP17, CYP19, CYP1B1, COMT and UGT1A1 affected the risk of prostate cancer in two different populations of African ancestry" does not provide any sense of positive or negative relationship which were labeled as "F". Hence, sentences labeled as "F" were eliminated from the test set.

Table 4.5: NB classifier prediction accuracy results - Cycle 2

| **Accuracy 0.8734525447042641** | | | | |
|---|---|---|---|---|
| | precision | recall | f1-score | support |
| N | 0.87 | 0.42 | 0.57 | 144 |
| Y | 0.87 | 0.98 | 0.93 | 583 |
| micro avg | 0.87 | 0.87 | 0.87 | 727 |
| macro avg | 0.87 | 0.70 | 0.75 | 727 |
| weighted avg | 0.87 | 0.87 | 0.86 | 727 |

Results of the prediction cycle 2, demonstrated an improved accuracy in all three models compared to the first prediction cycle. Within the cycle, SVM and LR models are performing well with competing accuracy. LSVM demonstrated an overall accuracy of 0.94. Precision and Recall measures for the positive relationships show 0.96 and 0.97 measures respectively. See table 4.6.

Table 4.6: LSVM classifier prediction accuracy results - Cycle 2

| **Accuracy 0.9436038514442916** | | | | |
|---|---|---|---|---|
| | precision | recall | f1-score | support |
| N | 0.88 | 0.83 | 0.85 | 144 |
| Y | 0.96 | 0.97 | 0.97 | 583 |
| micro avg | 0.94 | 0.94 | 0.94 | 727 |
| macro avg | 0.92 | 0.90 | 0.91 | 727 |
| weighted avg | 0.94 | 0.94 | 0.94 | 727 |

LR classifier found its accuracy score of 0.92 which is intermediate to NB and LSVM models. See table 4.7.

### 4.1.4 Cycle 3: Prediction accuracy results of newly trained models on PubMed data set

In the third cycle of prediction, the models were re-trained on the entire GAD data set. New models were used in the prediction task. This resulted in better prediction accuracy in all three models out of which LSVM and LR models lead. Accuracy scores were improved than both previous cycles.

Table 4.7: LR classifier prediction accuracy results - Cycle 2

| **Accuracy 0.922971114167813** | | | | |
|---|---|---|---|---|
| | precision | recall | f1-score | support |
| N | 0.78 | 0.85 | 0.81 | 144 |
| Y | 0.96 | 0.94 | 0.95 | 583 |
| micro avg | 0.92 | 0.92 | 0.92 | 727 |
| macro avg | 0.87 | 0.90 | 0.88 | 727 |
| weighted avg | 0.93 | 0.92 | 0.92 | 727 |

Table 4.8: NB classifier prediction accuracy results - Cycle 3

| **Accuracy 0.8762035763411279** | | | | |
|---|---|---|---|---|
| | precision | recall | f1-score | support |
| N | 0.85 | 0.46 | 0.59 | 144 |
| Y | 0.88 | 0.98 | 0.93 | 583 |
| micro avg | 0.88 | 0.88 | 0.88 | 727 |
| macro avg | 0.86 | 0.72 | 0.76 | 727 |
| weighted avg | 0.87 | 0.88 | 0.86 | 727 |

Table 4.9: LSVM classifier prediction accuracy results - Cycle 3

| **Accuracy 0.951856946354883** | | | | |
|---|---|---|---|---|
| | precision | recall | f1-score | support |
| N | 0.91 | 0.84 | 0.87 | 144 |
| Y | 0.96 | 0.98 | 0.97 | 583 |
| micro avg | 0.95 | 0.95 | 0.95 | 727 |
| macro avg | 0.94 | 0.91 | 0.92 | 727 |
| weighted avg | 0.95 | 0.95 | 0.95 | 727 |

Table 4.8, table 4.9, and table 4.10 show the prediction results using newly trained NB, LSVM, and LR models against the the PubMed data set. These models predict with an overall accuracy score of 0.87, 0.95 and 0.92 respectively marking the highest among the prediction cycles of the study.

Table 4.10: LR classifier prediction accuracy results - Cycle 3

| **Accuracy 0.9298486932599724** | | | | |
|---|---|---|---|---|
| | precision | recall | f1-score | support |
| N | 0.80 | 0.86 | 0.83 | 144 |
| Y | 0.97 | 0.95 | 0.96 | 583 |
| micro avg | 0.93 | 0.93 | 0.93 | 727 |
| macro avg | 0.88 | 0.90 | 0.89 | 727 |
| weighted avg | 0.93 | 0.93 | 0.93 | 727 |

## 4.2 Most important features learnt by the models

During the training process, models learn important features using the training data set and used the learned features in classifying unseen data provided. Therefore, it is important to look at the top features learnt by these models. According to the results, LSVM and LR models perform competitively. Hence, here we extracted the top 20 features for LSVM and LR models.

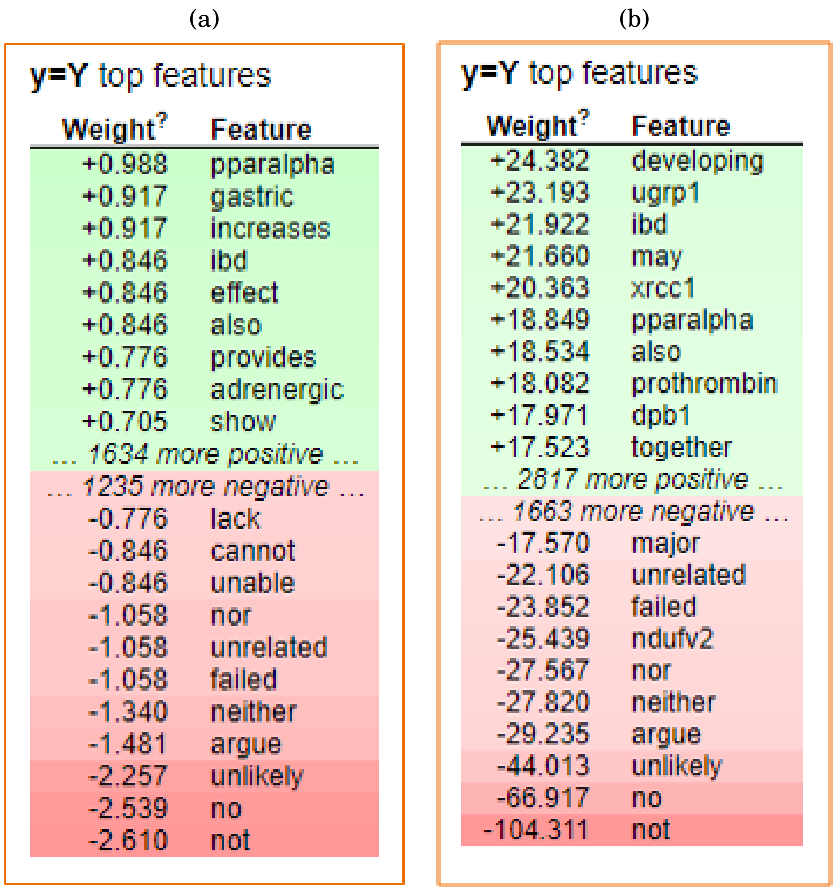
See figure 4.3 for the top most 20 features learnt by the LSVM model and LR model throughout the learning process. The green colour features represent the features learned for classifying unseen data as positive "P" class and the red colour features are used to classify data into negative "N" class.

## 4.3 Presenting results in useful form

Presenting the final results of the third cycle of the study involved in two different forms. The first one is to save results in a database and let the users query from the findings to shortlist interested positive associations in a through a web based interface.

### 4.3.1 Php interface to query the predicted results

Findings were converted into MySQL format and imported to a MySQL database. Developed a web-based php interface with Codeigniter framework and ajax. It supports querying the results based on genes or disease names. The resulting records



(a)

y=Y top features

| Weight? | Feature |
|---|---|
| +0.988 | pparalpha |
| +0.917 | gastric |
| +0.917 | increases |
| +0.846 | ibd |
| +0.846 | effect |
| +0.846 | also |
| +0.776 | provides |
| +0.776 | adrenergic |
| +0.705 | show |
| … 1634 more positive … | |
| … 1235 more negative … | |
| -0.776 | lack |
| -0.846 | cannot |
| -0.846 | unable |
| -1.058 | nor |
| -1.058 | unrelated |
| -1.058 | failed |
| -1.340 | neither |
| -1.481 | argue |
| -2.257 | unlikely |
| -2.539 | no |
| -2.610 | not |

(b)

y=Y top features

| Weight? | Feature |
|---|---|
| +24.382 | developing |
| +23.193 | ugrp1 |
| +21.922 | ibd |
| +21.660 | may |
| +20.363 | xrcc1 |
| +18.849 | pparalpha |
| +18.534 | also |
| +18.082 | prothrombin |
| +17.971 | dpb1 |
| +17.523 | together |
| … 2817 more positive … | |
| … 1663 more negative … | |
| -17.570 | major |
| -22.106 | unrelated |
| -23.852 | failed |
| -25.439 | ndufv2 |
| -27.567 | nor |
| -27.820 | neither |
| -29.235 | argue |
| -44.013 | unlikely |
| -66.917 | no |
| -104.311 | not |

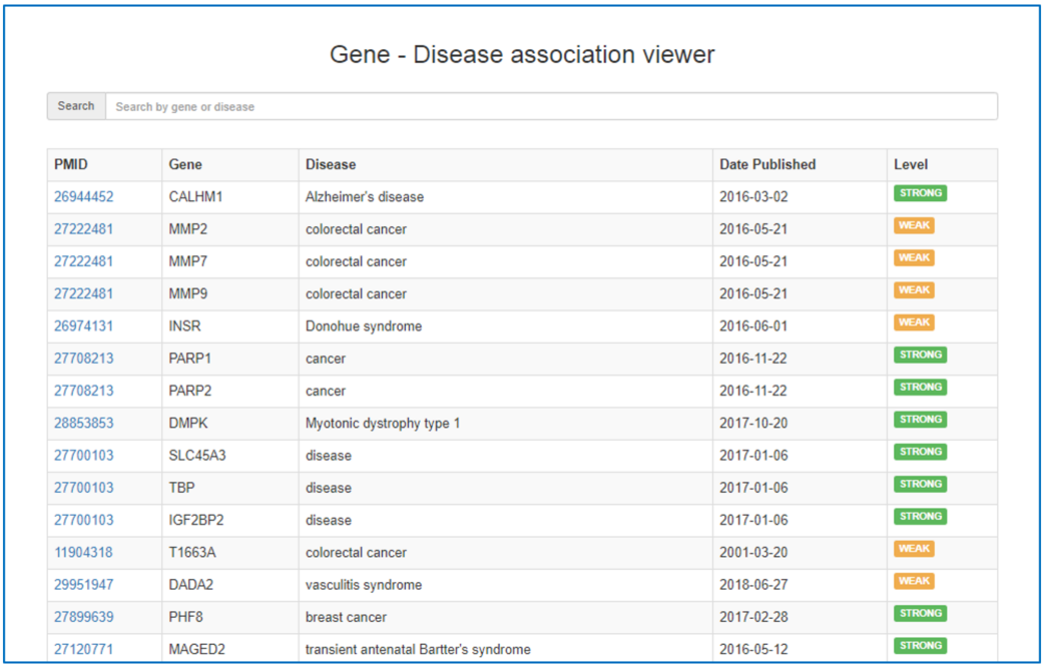
FIGURE 4.3. (a) Top 20 features of LSVM model. (b) Top 20 features of LR model

contain PMID (Pubmed ID), HGNC symbol, Disease name, Date of publication and the degree of the relationship (whether it is a strong or weak relationship). Figure 4.4 illustrates the interface that provide querying facility.

For example, if the query is about "colorectal cancer", the interface will list down all the related gene entities with the aforementioned information like PMID, Date published and the degree of association. See figure 4.5

### 4.3.2 Visualize predictions as a network graph

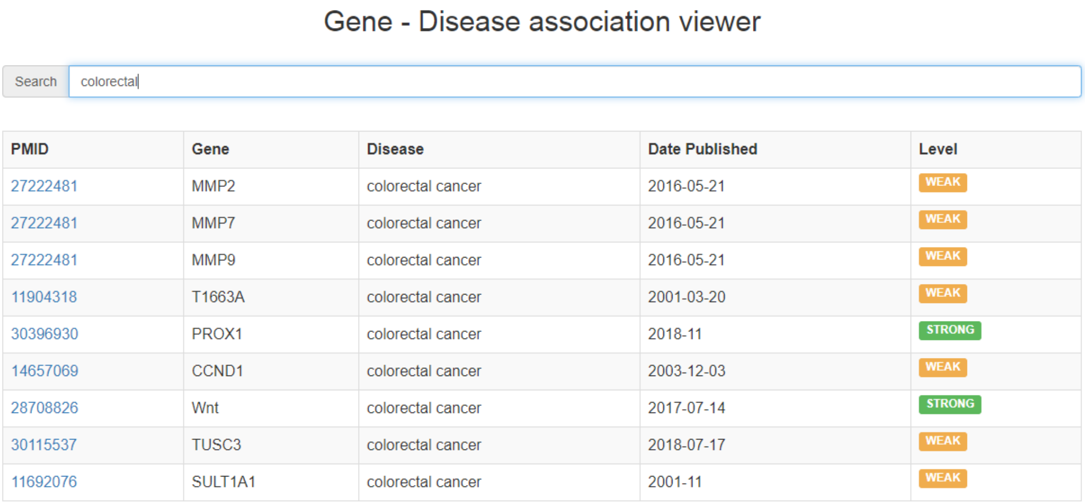
In this regard, the results were loaded in to a Pandas data frame first. Then using NetworkX python library the edge list and nodes prepared. Using Matplotlib 2D plotting library, a network diagram was drawn. Using this network graph representation, one can see the multidimensional relationships clearly. Figure 4.6 show a portion of the network graph based on the true positive predictions.



Gene - Disease association viewer

| PMID | Gene | Disease | Date Published | Level |
|---|---|---|---|---|
| 26944452 | CALHM1 | Alzheimer's disease | 2016-03-02 | STRONG |
| 27222481 | MMP2 | colorectal cancer | 2016-05-21 | WEAK |
| 27222481 | MMP7 | colorectal cancer | 2016-05-21 | WEAK |
| 27222481 | MMP9 | colorectal cancer | 2016-05-21 | WEAK |
| 26974131 | INSR | Donohue syndrome | 2016-06-01 | WEAK |
| 27708213 | PARP1 | cancer | 2016-11-22 | STRONG |
| 27708213 | PARP2 | cancer | 2016-11-22 | STRONG |
| 28853853 | DMPK | Myotonic dystrophy type 1 | 2017-10-20 | STRONG |
| 27700103 | SLC45A3 | disease | 2017-01-06 | STRONG |
| 27700103 | TBP | disease | 2017-01-06 | STRONG |
| 27700103 | IGF2BP2 | disease | 2017-01-06 | STRONG |
| 11904318 | T1663A | colorectal cancer | 2001-03-20 | WEAK |
| 29951947 | DADA2 | vasculitis syndrome | 2018-06-27 | WEAK |
| 27899639 | PHF8 | breast cancer | 2017-02-28 | STRONG |
| 27120771 | MAGED2 | transient antenatal Bartter's syndrome | 2016-05-12 | STRONG |

FIGURE 4.4. Query interface developed using php MySQL



Gene - Disease association viewer

| PMID | Gene | Disease | Date Published | Level |
|---|---|---|---|---|
| 27222481 | MMP2 | colorectal cancer | 2016-05-21 | WEAK |
| 27222481 | MMP7 | colorectal cancer | 2016-05-21 | WEAK |
| 27222481 | MMP9 | colorectal cancer | 2016-05-21 | WEAK |
| 11904318 | T1663A | colorectal cancer | 2001-03-20 | WEAK |
| 30396930 | PROX1 | colorectal cancer | 2018-11 | STRONG |
| 14657069 | CCND1 | colorectal cancer | 2003-12-03 | WEAK |
| 28708826 | Wnt | colorectal cancer | 2017-07-14 | STRONG |
| 30115537 | TUSC3 | colorectal cancer | 2018-07-17 | WEAK |
| 11692076 | SULT1A1 | colorectal cancer | 2001-11 | WEAK |

FIGURE 4.5. Result for the query "Colorectal cancer"

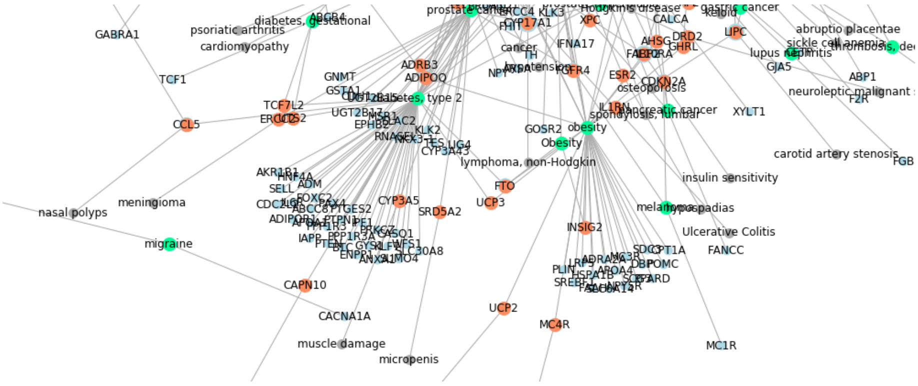


FIGURE 4.6. A segmant of the graph representation of the results

## 4.4 Evaluation metrics

The gene-disease relations type (Positive or Negative) extraction systems performance measuring matrix composed of three performance measures. They are, Precision (P), Recall (R) and F-measure (F) scores. These measures can be used to evaluate the association/relations type extraction accuracy of the proposed system. The matrix identifies two types of errors in the results generated by the proposed system, which will help us identifying the performance level of the underlying implementation of the gene-disease relations extraction model. The two types of errors are type I given by False Positives (FP) and type II given by False Negatives (FN) that could be identified by Precision and Recall respectively. F-score provides an overall performance score considering both Precision and Recall scores. F-score is defined as the harmonic mean of the Precision and Recall.

Precision is calculated based on the following equation. TP over the combination of TP and FP gives the Precision score.

(01 - Precision)

$$P = \frac{TP}{TP + FP}$$

Recall is calculated based on the following equation. TP over the combination of TP and FN gives the Recall score.

(02 - Recall)

$$R = \frac{TP}{TP + FN}$$

The harmonic mean is calculated as follows to calculate the F-score as given in below.

(03 - F-Score)

$$F = \frac{2 * P * R}{P + R}$$

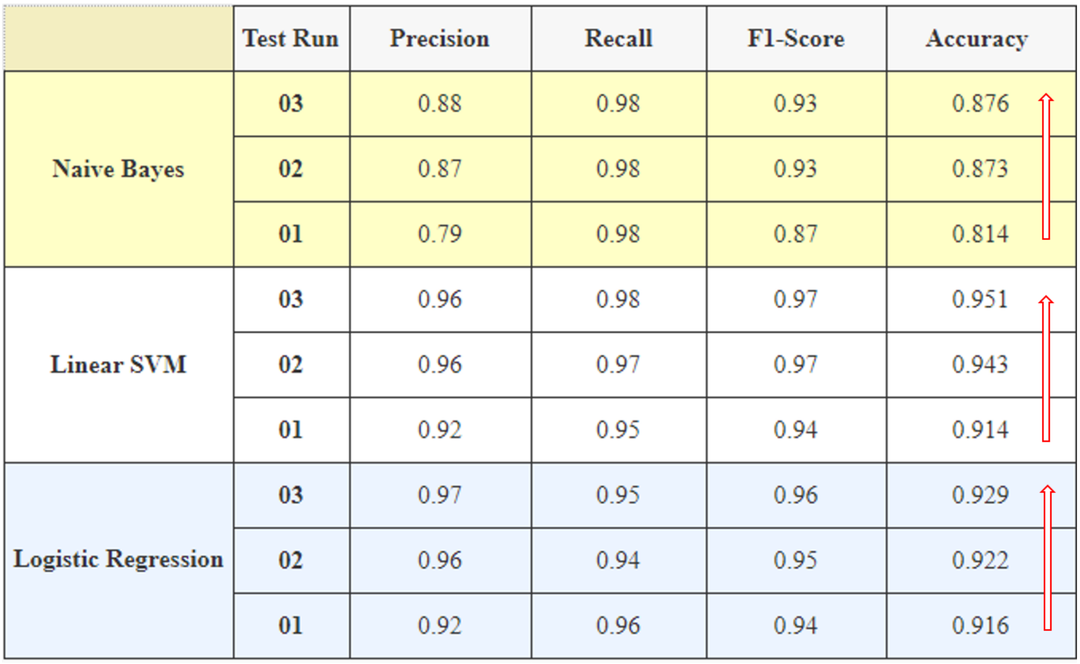
## 4.5 Evaluation plan

As described earlier, evaluation of the models was done against two data sets. The first one is the GAD data set from which the models are trained. In this evaluation a 30% segment of the data set was considered for the purpose of evaluating the predictive models. the results are explained under section 4.1.1.

The next evaluation was done based on a totally different data set which has been manually prepared and labeled by the author based on PubMed abstracts that are freely available. This data set consists of 1000 records from which 727 data records used for the evaluation of prediction accuracy of the models. For the results obtained, see sections 4.1.3 and 4.1.4.

## 4.6 Evaluation of results

Figure 4.7 summarizes results from all three testing cycles. It is evident that the accuracy column shows a clear gradual improvement in prediction accuracy in each model. Out of all three models LSVM performs better in prediction task according to the results obtained. Secondly, LR model performs better. 01 - Precision, 02 - Recall and 03 - F-Score were used to compare models' prediction accuracy.



| | Test Run | Precision | Recall | F1-Score | Accuracy |
|---|---|---|---|---|---|
| Naive Bayes | 03 | 0.88 | 0.98 | 0.93 | 0.876 |
| | 02 | 0.87 | 0.98 | 0.93 | 0.873 |
| | 01 | 0.79 | 0.98 | 0.87 | 0.814 |
| Linear SVM | 03 | 0.96 | 0.98 | 0.97 | 0.951 |
| | 02 | 0.96 | 0.97 | 0.97 | 0.943 |
| | 01 | 0.92 | 0.95 | 0.94 | 0.914 |
| Logistic Regression | 03 | 0.97 | 0.95 | 0.96 | 0.929 |
| | 02 | 0.96 | 0.94 | 0.95 | 0.922 |
| | 01 | 0.92 | 0.96 | 0.94 | 0.916 |

FIGURE 4.7. Results of all three testing cycles

According to the final prediction accuracy results, confusion matrix for each model was generated. Confusion matrices provide sound basis for evaluating the model's performance in a number of aspects. As genes and disease relations predicted by the system are related to the bio-medical domain, it is more important to focus on Type-II error rather than Type-I errors in predictions. In bio-medical domain Type-1 error impact is lower than that of Type-II error. For example, predicting a disease condition as positive where the patient is actually not having the disease falls under Type-I error. On the other hand, predicting a disease condition as negative where the patient is actually having the disease falls under Type-II error. Therefore minimizing the Type-II error is more important in predictions.

Further, two measures calculated to assess the suitability of the models. They are Sensitivity and Specificity characteristics.

- **Sensitivity :** Measures how often a test correctly generates a positive result (True Positive Rate)

- **Specificity :** Measures a test's ability to correctly generate a negative result (True Negative Rate)

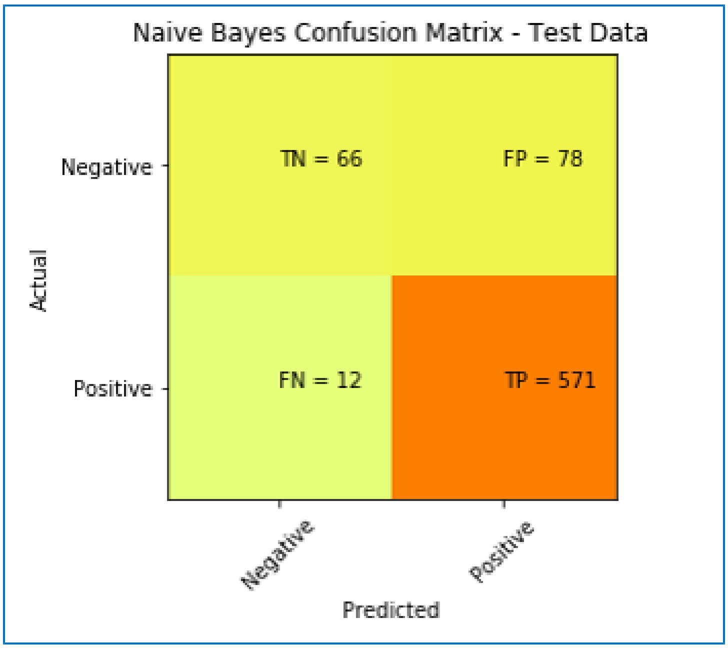


FIGURE 4.8. Confusion matrix for NB model predictions

Figure 4.8 illustrates the confusion matrix for NB model predictions. There, False Negative count is 12 which is a good characteristic as it reduces the Type-II error of the model.

Specificity = TN/(FP + TN) = 66/(78 + 66) = 0.45

Sensitivity = TP/(TP+FN) = 571/(571+12) = 0.98

High sensitivity means that the model model is correctly generating a positive result with a 98% accuracy. Which is a really good sign. Even the model with lower accuracy demonstrated better in predicting positive relations.

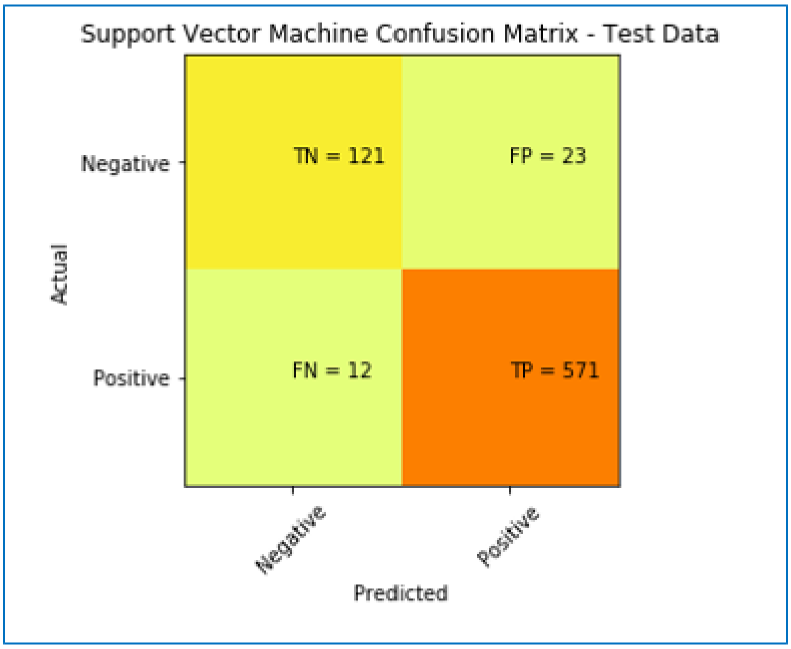


FIGURE 4.9. Confusion matrix for LSVM model predictions

Figure 4.9 illustrates the confusion matrix for LSVM model predictions. There, False Negative count remains as 12 which is a good characteristic. This is exactly similar to that of the NB. Type-II error of the model is very small compared to the data set.

Specificity = TN/(FP + TN) = 121/(23 + 121) = 0.84

Sensitivity = TP/(TP+FN) = 571/(571+12) = 0.98

Apart from the sensitivity being same as the NB model, LSVM has an improved figure in specificity as well. LSVM is better than the NB according to the results.

Figure 4.10 illustrates the confusion matrix for LR model predictions. There, False Negative count increases up to 31 which is not a good characteristic. Type-II error of the model is bigger than the previous models. Even though the LR model performs better in overall accuracy, when looking at the Type-II error which has a higher significance in bio-medical domain, this model could generate erroneous results.

Specificity = TN/(FP + TN) = 124/(20 + 124) = 0.86

Sensitivity = TP/(TP+FN) = 552/(552+31)

Sensitivity is lower than the LSVM model and NB model, but specificity remains higher than the NB model.

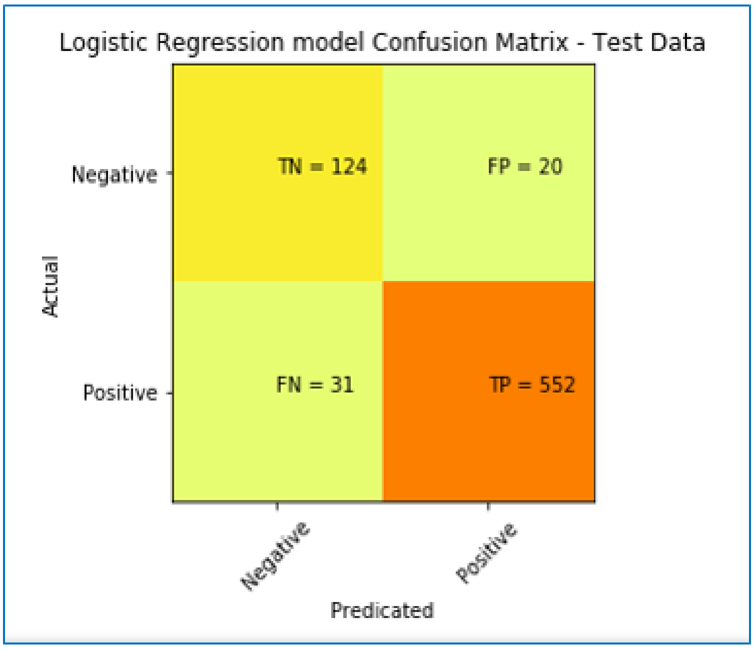


FIGURE 4.10. Confusion matrix for LR model predictions

In conclusion, the LSVM model performs better out of all three model trained and validated based on two data sets.

# Chapter 5

# Future work and conclusion

## 5.1 Future work

In this study, main focus is on sentence level gene-disease association type prediction using a supervised approach. Within the limited time duration, a new data set has been labeled based on manually identified gene-disease association type (positive or negative association) of thousand sentences extracted from freely accessible PubMed abstracts. The potential of preparing a large data set is promising in this approach as there is a huge pool of sentences with gene disease mentions available freely through this database. One of the future avenues is to expand the data set curated from PubMed data base abstracts for future text mining and machine learning research.

This research work was only limited to cancer type diseases. There are multiple categories of diseases available. Further research can carry out based on expanded version of disease categories to evaluate the accuracy of predictions. For example, Genetic Disease, Hematologic Disease, Gastrointestinal Disease, Immunodeficiency Disease, Infectious Disease, Allergic Disease, Autoimmune Disease, etc.

Further, the study can be expanded to unsupervised domain of learning. This would require a larger robust data set.

The 2D graph representation can be converted in to a 3D space network graph representation and allowing users to query based on the nodes which will zoom in the relations associated with the particular node. Hence, finding better presentation structures of the results would be considered as another future work.

## 5.2 Conclusion

In conclusion, predicting the gene-disease association type that is whether the gene mentioned in sentence having an association with the disease mentioned in the sentence and the gene mentioned in the sentence have no association with the disease mention can be successfully predicted using machine learning models. The context used for this research study is PubMed abstracts. Models were trained using the sentences appeared in conclusions of the papers published in the GAD database. Even the models trained on GAD records which were not derived and labeled based on the abstract sections, were able to predict abstract sentences association types with a good accuracy.

Among the models trained (NB, LSVM, and LR), LSVM proved to be the best after evaluating the results generated with an overall accuracy of 0.95. The predicted results were further processed for proper representation based on a php based querying interface and a network graph representation for visualizing multidimensional relatiionships among entities of interest.

# References

[1] P. Larrañaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. A. Lozano, R. Armañanzas, G. Santafé, A. Pérez, and V. Robles, “Machine learning in bioinformatics,” *Briefings in Bioinformatics*, vol. 7, no. 1, pp. 86–112, 2006.

[2] A. Manconi, E. Vargiu, G. Armano, and L. Milanesi, “Literature retrieval and mining in bioinformatics: State of the art and challenges,” *Advances in Bioinformatics*, vol. 2012, 2012.

[3] L. J. Jensen, J. Saric, and P. Bork, “Literature mining for the biologist: From information retrieval to biological discovery,” *Nature Reviews Genetics*, vol. 7, no. 2, pp. 119–129, 2006.

[4] Q.-C. Bui, “Relation Extraction Methods for Biomedical Literature Relation Extraction Methods for Biomedical Literature Thesis,” no. september, 2012.

[5] U. Hahn and J. Wermter, “Levels of natural language processing for text mining,” *Text Mining for Biology and Biomedicine*, pp. 13–41, 2006. [Online]. Available: http://books.google.com/books?id=xkNRAAAAMAAJ&pgis=1

[6] Y. Garten, A. Coulet, and R. B. Altman, “Recent progress in automatically extracting information from the pharmacogenomic literature,” *Pharmacogenomics*, vol. 11, no. 10, pp. 1467–1489, 2010.

[7] T. H. Nguyen and R. Grishman, “Relation Extraction: Perspective from Convolutional Neural Networks,” *Workshop on Vector Modeling for NLP*, pp. 39–48, 2015.

[8] K. Lee, B. Kim, Y. Choi, S. Kim, W. Shin, S. Lee, S. Park, S. Kim, A. C. Tan, and J. Kang, “Deep learning of mutation-gene-drug relations from the literature,” *BMC Bioinformatics*, vol. 19, no. 1, pp. 1–13, 2018.

[9] D. S. Miranda, "Automated Detection of Adverse Drug Reactions in the Biomedical Literature Using Convolutional Neural Networks and Biomedical Word Embeddings," 2018. [Online]. Available: http://arxiv.org/abs/1804.09148

[10] T. Huynh, Y. He, A. Willis, and S. Uger, "Adverse Drug Reaction Classification With Deep Neural Networks," in *COLING*, 2016.

[11] H. Gurulingappa, A. M. Rajput, A. Roberts, J. Fluck, M. Hofmann-Apitius, and L. Toldo, "Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports," *Journal of Biomedical Informatics*, vol. 45, no. 5, pp. 885–892, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.jbi.2012.04.008

[12] J. Zhou and B. q. Fu, "The research on gene-disease association based on text-mining of PubMed," *BMC Bioinformatics*, vol. 19, no. 1, pp. 1–8, 2018.

[13] S. Lee, D. Kim, K. Lee, J. Choi, S. Kim, M. Jeon, S. Lim, D. Choi, S. Kim, A. C. Tan, and J. Kang, "BEST: Next-generation biomedical entity search tool for knowledge discovery from biomedical literature," *PLoS ONE*, vol. 11, no. 10, pp. 1–16, 2016.

[14] K. Lee, W. Shin, B. Kim, S. Lee, Y. Choi, S. Kim, M. Jeon, A. C. Tan, and J. Kang, "HiPub: Translating PubMed and PMC texts to networks for knowledge discovery," *Bioinformatics*, vol. 32, no. 18, pp. 2886–2888, 2016.

[15] S. Pletscher-Frankild, A. Pallejà, K. Tsafou, J. X. Binder, and L. J. Jensen, "DISEASES: Text mining and data integration of disease-gene associations," *Methods*, vol. 74, pp. 83–89, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.ymeth.2014.11.020

[16] M. Song, W. C. Kim, D. Lee, G. E. Heo, and K. Y. Kang, "PKDE4J: Entity and relation extraction for public knowledge discovery," *Journal of Biomedical Informatics*, vol. 57, pp. 320–332, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.jbi.2015.08.008

[17] H. Zhou and J. Skolnick, "Systems Biology A knowledge-based approach for predicting gene-disease associations," 2016.

[18] A. P. Davis, C. J. Grondin, R. J. Johnson, D. Sciaky, R. McMorran, J. Wiegers, T. C. Wiegers, and C. J. Mattingly, "The Comparative Toxicogenomics Database: update 2019," *Nucleic Acids Research*, vol. 47, no. D1, pp. D948–D954, 09 2018. [Online]. Available: https://doi.org/10.1093/nar/gky868

# Appendix A

## Downloading gene symbols list from Ensemble

```
//Installing biomaRt package
source("http://bioconductor.org/biocLite.R")
biocLite("biomaRt")

//Add library
library(biomaRt)

//View data sets available in ensembl database
ensembl=useMart("ensembl")
listDatasets(ensembl)

//Creating a data set object
ensembl_mart <- useMart("ensembl", dataset="hsapiens_gene_ensembl")

//verify data set object
ensembl_mart

//list all the attributes available for the data set
listAttributes(ensembl_mart)

//Retrieving all the gene HGNC symbols of the chromosome 1
chromosome1 = getBM(attributes = (
"hgnc_symbol"), filters = "chromosome_name", values = "1", mart = ensembl_mart)
> chromosome1
```

### Selecting 500 sample sentences

```
import random

def reservoir_sampling(l, k):
    iterator = iter(l)
    try:
        result = [next(iterator) for _ in xrange(k)]
    except StopIteration:
        raise ValueError("sample limit is too large")

    for i, item in enumerate(iterator, start=k):
        number = random.randint(0, i)
        if number < k:
            result[number] = item

    random.shuffle(result)
    return result

with open('./output/both_included.csv') as population_file,
open('./output/filtered_samle.csv', 'a') as sample_file:
    for line in reservoir_sampling(population_file, 500):
        sample_file.write(line)
```

## Processing gene symbols in to one file

```
import os
import glob
import pandas as pd

# path to the directory that holds all 24 gene files
os.chdir("./output_human_genes")

# extention format for files is ".csv"
extension = 'csv'

# read all file names in to a list
all_filenames = [i for i in glob.glob('*.{}'.format(extension))]

# combine all files in the list using pandas "concat()" method
combined = pd.concat([pd.read_csv(f) for f in all_filenames])
# write the concatenated gene symbols into "combined_genes.csv"
combined.to_csv("combined_genes.csv",index=False,encoding='utf8')
```

## Processing raw diseases list

```
import os
import pandas as pd
import nltk
from nltk import word_tokenize

with open('./cancer_diseases.txt', "r") as line:
        with open('./Final_Datasets/diseases.csv', "w") as diseases:
            for disease in line:
                disease_list = nltk.word_tokenize(disease.translate(
                    string.maketrans("",""), string.punctuation))
                diseases.writelines(
                    "{}\n".format(x) for x in disease_list)

csv_input_file = './Final_Datasets/diseases.csv'
data = pd.read_csv(csv_input_file, sep='\t', encoding='utf8')
data.drop_duplicates([data.columns[0]],keep='first',inplace = True)
data['column2'] = map(lambda x: x.lower(), data['column1'])
data = data.drop(['column1'],axis=1)
data = data.sort_values(['column2'])
data.to_csv("./diseases_cancer.csv",sep=',',index=False,encoding='utf8')
```

## Preparing disease and genes lists

```
# the list to store HGNC gene symbols
geneTargets = []
# genes list based on the human genes
with open("./genes/all_combined_genes.csv", "r") as ins:
    for target in ins:
        # removes any trailing or leading characters
        # and appends to the list
        geneTarget = target.strip()
        geneTargets.append(geneTarget)

# the list to store diseases
diseaseTargets = []
# cancer types list based on National Cancer Institute
# site listing (url:https://www.cancer.gov/types)
with open("./diseases/cancer_diseases_list.csv", "r") as ins:
    for target in ins:
        # removes any trailing or leading characters
        # and appends to the list
        diseaseTarget = target.strip()
        diseaseTargets.append(diseaseTarget)
```