EVIDENCE BASED MEDICAL QUERY SYSTEM ON LARGE SCALE DATA

A THESIS IN

Computer Science

Presented to the Faculty of the University
Of Missouri-Kansas City in partial fulfillment
Of the requirements for the degree

MASTER OF SCIENCE

By

VENKATA PRAMOD GUPTA BAVIRISETTY
B.Tech, National Institute of Technology Warangal, 2011

Kansas City, Missouri
2014

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Computing and Engineering, have examined a thesis titled "Evidence based Medical Query System on Large Scale Data" presented by Venkata Pramod Gupta Bavirisetty, candidate for the Master of Science degree, and hereby certify that in their opinion, it is worthy of acceptance.

Supervisory Committee

Yugyung Lee, Ph.D., Committee Chair
School of Computing and Engineering

Yongjie Zheng, Ph.D., Committee
School of Computing and Engineering

Praveen Rao, Ph.D., Committee
School of Computing and Engineering

EVIDENCE BASED MEDICAL QUERY SYSTEM ON LARGE SCALE DATA

Venkata Pramod Gupta Bavirisetty, Candidate for the Master of Science Degree

University of Missouri-Kansas City, 2014

ABSTRACT

As huge amounts of data are created rapidly, the demand for the integration and analysis of such data has been growing steadily. It is especially essential to retrieve relevant and accurate evidence in healthcare and biomedical research. Even though query systems based on Ontology, Medical Subject Headings (MeSH), or keyword searches are available, query systems based on evidence and effective retrieval of data from large collections of clinical data are not sufficiently available.

This thesis proposes a novel approach to analyze big data sets collected from Clinical trials research and discover significant evidence and association patterns with respect to conditions, treatment, and medication side effects. Our approach makes use of machine learning techniques in the Apache Hadoop framework with support from MetaMap and RxNorm. In this thesis, a heuristic measure of empirical evidence was newly designed considering the association degree of conditions, treatment, and medication side effects and the percentage of people affected. The Apriori algorithm was used to discover strong positive association rules with various measures including support, and confidence. We have examined a large and complex data set (12,327 study results) from clinicaltrials.gov and identified 8,291 strong association rules and 59,228 combinations with 432,841 subjects, 1761 conditions, 2836 drugs, and 27 side effects.

The significance of these association patterns was evaluated in terms of the impact factor representing the percentage of the population with a high rate of side effects. Using these association rules and combination strengths, an evidence based query system was implemented to answer some integral questions. This query system also provided an interface to retrieve relevant publications from PubMed. The searching outcomes from this query system are compared with those from the PubMed search based on medical subject headings.

Table of Contents

ILLUSTRATIONS

Figure Page

TABLES

Table Page

CHAPTER 1

INTRODUCTION

1.1 Motivation

In the medical domain information plays a key role. Over the internet huge amount of medical data is created daily and is available freely. There has been huge amount of medical research data added to the medical repositories like PubMed [22], Clinical Trials [4], Medline plus [17], etc. For effective retrieval of the data, there is a need for a query system that searches through the medical data available and presents the relevant results efficiently. A general web search engine tries to serve the information by retrieving and ranking of user’s query, which has a huge impact on how data is represented and organized. This kind of searching causes a serious problem while the task is in medical domain.

However later new medical query systems were built on medical repositories like PubMed [22], Medline Plus [17], Cochrane [5], which provides comprehensive coverage of evidence based medical literature. PubMed system mentioned above uses the MeSH term for retrieving the literatures. This system provides all the literatures with the matching MeSH term in the repository. There is another query system, based on text summarization techniques that makes use of the Unified medical language system [28], an ontology knowledge source from National Library of Medicine [20]. This system queries the data based on the ontologies of the medical terms and returns the literature based on ontology mapping.

In order to keep updated with the latest increase in the ongoing research in the medical area the practitioners needs an efficient way to retrieve data related to a particular domain in the medical field from the datasets. The retrieved results by the query system will also need to be reliable and effective, so that this can be linked with the online literatures to get the relevant online medical evidence. As the practitioners deals with variety of areas such as diagnosis, treatment, therapy, etiology the linked literature should be contextually relevant and this would provide potential benefits to the Health Practitioners.

1.2 Problem Statement

The medical query systems available online provides an evidence based query system on the online medical literatures available by making use of the MeSH term or the ontologies related to the medical search term. Due to the growth of medical data over the internet, the effective retrieval of data from the data set is becoming very expensive and time consuming. Due to this, the support for query systems on large volumes of scientific data is becoming increasingly important.

The queries in the current systems were based on ontologies or Mesh terms. These applications does not provide results based on the side effects or the percentage of people effected or the test groups that were considered at the time of research. Using this information for querying would facilitate the practitioners with much relevant and effective medical literature.

1.3 Thesis Outline

The thesis focuses on two important problems that usually occur at the time of querying the medical data repositories. The first problem is the big data challenge which requires high computational resources to query effectively and efficiently. The second problem is the related literature being resulted from the query system using the evidence available. All the query systems available online results the queries based on the either keyword or Ontologies or MeSH term but not on the side effects or the percentage of people effected or any other evidences.

For the thesis, we have considered Clinical Trials [4] as the data set. This data compromises of medical studies on conditions, studies with results and proposals on medical conditions and use of combination of drugs for a condition and the side effects and the medical groups used for these studies. This also compromises of the statistics like the group size, group properties, drugs, time period etc. We have considered 12,327 research studies for the thesis.

In this thesis we present our work towards building a high performance data analysis on distributed system using MapReduce. The system processes the clinical data and pull the required data from the entire dataset. For high performance querying, the data from the distributed system is loaded into a centralized system and a PHP query system is built on top of it. We have observed 1761 unique Medical conditions and 2836 unique drugs used in the entire studies.

In order to provide the results based on the evidence, we use the condition, drug and side effect combination and the frequency of this combination in the overall data set and the percentage of people effected with this side effect in the overall data set. Along with the above method, we used association rule mining on the data to find the relevant association rules.

CHAPTER 2

RELATED WORK

2.1 Overview

In this chapter, we will discuss about the Existing Medical Query Systems, Existing Large Scale based Query Systems. We will go through the query criteria's of the existing systems and compare them with the model that is presented in the thesis.

2.2 Problems of Existing Medical Query Systems

Most of the current search engines only perform a shallow string processing because of the lack of understanding of natural languages and human intelligence which leaves the users the users to go through pages of results to find the relevant literatures. Usage of Ontologies provided much better results by using the ontologies, related to the users search query. The documents are summarized using the taxonomies which lets it retrieval much faster. But most of these search engines does not make use of the evidences provided from the research documents which could be helpful in much better retrieval of the relevant information.

The amount of medical information available online is growing rapidly and this leaves us with the Big Data issue. Most of the medical query systems available online does not make use of the Large scale framework for processing the information.

2.3 Existing Medical Query Systems

Medical query systems allow us to search and retrieve information from the medical repositories. Existing medical query systems use different methods for searching the repositories such as Keyword or Ontology or Mesh terms. Some of the Medical Systems use either Keyword or Ontology or Mesh terms to search the datasets.

Search by Keyword:

This is the most traditional search where the user can search medical content by using keywords, the search engine returns the documents which has those keywords in the document used by the user while querying [16]. These query systems usually store the medical information in the relational database [3] and search the database using the keywords and retrieves the related medical documents. Some query systems usually search the medical documents for the keywords and retrieve the documents matching the keyword criteria. Usually the search engines rank the documents according to the user's query and organize the data and present it to the user.

Search by Ontology:

Many applications use ontology [7] for querying medical repositories. Usually most of the query systems use some of the online available medical ontology repositories to get all the information related to the medical terminology. Ontologies are descriptions of the concepts and relationships. Most of the repositories use either Unified Medical Language system (UMLS) [28] or National Library of Medicine (NLM) [20]. Usually some query systems [21] revise the users search query with the ontology knowledge and adds the relevant terms. Then distance of each sentence in the document is calculated. If the distance is less than the threshold then they are included in the output.

Some other algorithms use the ontologies for getting the related terminology for the keyword and the use all the keywords for lexical searching [16]. Here all the keywords collected from the ontology repository are searched in the document. Then the documents with most keywords available are retrieved and presented the user.

Search by MeSH Term:

This is the kind of search used by the PubMed [22] for retrieving the medical articles. Initially all the articles inserted into the PubMed are examined, and the most specific MeSH (Medical Subject Headings) [22] terms with a related heading and sub heading, typically ten to twelve are assigned to each citation.

Each MeSH term is added into the database for each citation and the indexes are updated every week for all the citations in the repository. MeSH terms are arranged hierarchically by subject categories with more specific terms arranged beneath broader terms. Once a search is done PubMed automatically explodes the search by including all the narrower terms. These MeSH terms are searched against the database consisting of the MeSH terms related to the each citation and the related citations were returned to the user.

Table 1: Comparison between Medical Query Systems

Query System	Query Type	Ontology Source	Large Scale Framework	Output Criteria	Machine Learning Algorithm
High Performance Spatial Query System[1.]	Spatial Objects based Queries	No	Hadoop	-	None
Query Expansion Framework System[34]	Ontology based	NLM	No	Concept threshold	None
MedSearch[24]	Semantics based	NLM	No	-	None
PubMed[13]	MeSH term based	MeSH	No	No. of MeSH terms	None
Language Query Patterns Query system[9]	Keyword based	UMLS	No	Keyword Ranking	None
Medical Information Summarization System[1]	Ontology based	UMLS	No	Distance based Keyword ranking	None
Evidence based Query model	Keyword & MetaMap	MetaMap	Hadoop	Combination Strength	Apriori Rule Mining

2.4 Existing Large Scale based Query Systems

Support for high performance queries on large volumes of medical data is becoming increasingly important in many applications. The major requirement for such a query model is effective querying of large amounts of data with fast responses, which is faced with two major challenges: the “Big data” challenge and high computational complexity. Some of the query systems available [1] makes use of the MapReduce [13] paradigm for querying the large amount of data. This query system uses spatial queries to query the data. Usually the method followed consists of the following steps: Object Segmentation, Region Segmentation, Feature extraction and Data Management and Queries. Usually these systems make use of the Hadoop Environment [13] and MapReduce paradigm to process the large amounts of data.

2.5 Summary

As we have discussed above, all the medical query systems available online are not compatible for large scale data. Most of the medical query systems either use keyword or ontology or MeSH terms for querying the systems but none of the query systems use evidence from research studies for querying the data.

CHAPTER 3

EVIDENCE BASED MEDICAL QUERY MODEL

3.1 Overview

In this chapter, we will discuss about the Evidence based medical query model. This includes collecting data from the Clinical trials site [4] and loading the data into the Hadoop cluster [13] that has been setup in the local for filtering the data in parallel using the MapReduce Paradigm.

The data obtained from the Clinical Trials [4] is stored is in the XML format. The data consists of all the information related to the research done by the teams on the test subjects. The data is filtered and linked with MetaMap [8] for collecting the related medical terms and Meta Score. For each combination of Condition, Drug and Side effect the percentage of people affected and the frequency of the combination in the entire data set is calculated. Using these two parameters the strength of the combination is calculated and is used to order the data returned by querying the data model. Apriori association rule mining algorithm is used on the data retrieved from the clinical trials and association rules were generated.

The resulted data from the model is used to link with PubMed [22] to pull the related publications containing all the variables in the combination. The data model can also be used as a simple query system and link the results with RxNorm[24] to pull the related drug details.

3.2 System Architecture

The architecture consists of a query interface where the user can input the query term onto the model. The webserver consists of the data model which is loaded from the Hadoop environment [13]. The data analysis and MetaMap linking were explained in the architecture diagram. The architecture diagram shows how the query is sent to the webserver and linked with PubMed [22] and RxNorm [24]. Here in the figure all the orange color represents the models used or implemented within the system where as the sky blue represents the external models.

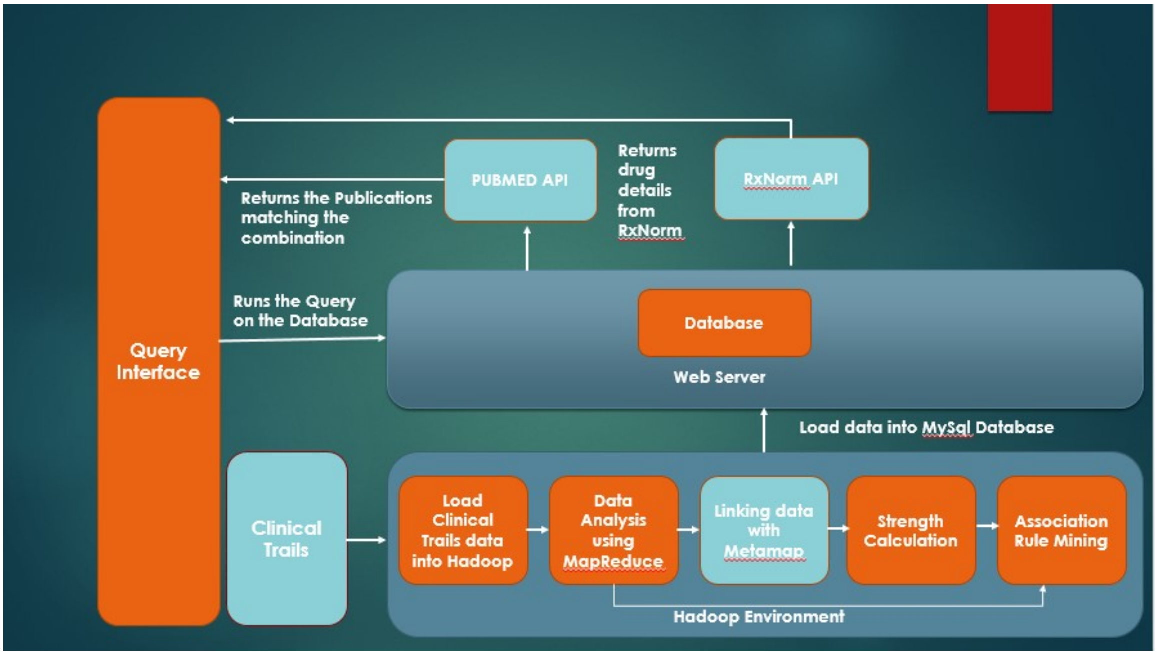

Query Interface
PUBMED API
Returns the Publications matching the combination
Returns drug details from RxNorm
RxNorm API
Runs the Query on the Database
Database
Web Server
Load data into MySql Database
Clinical Trails
Load Clinical Trails data into Hadoop
Data Analysis using MapReduce
Linking data with Metamap
Strength Calculation
Association Rule Mining
Hadoop Environment

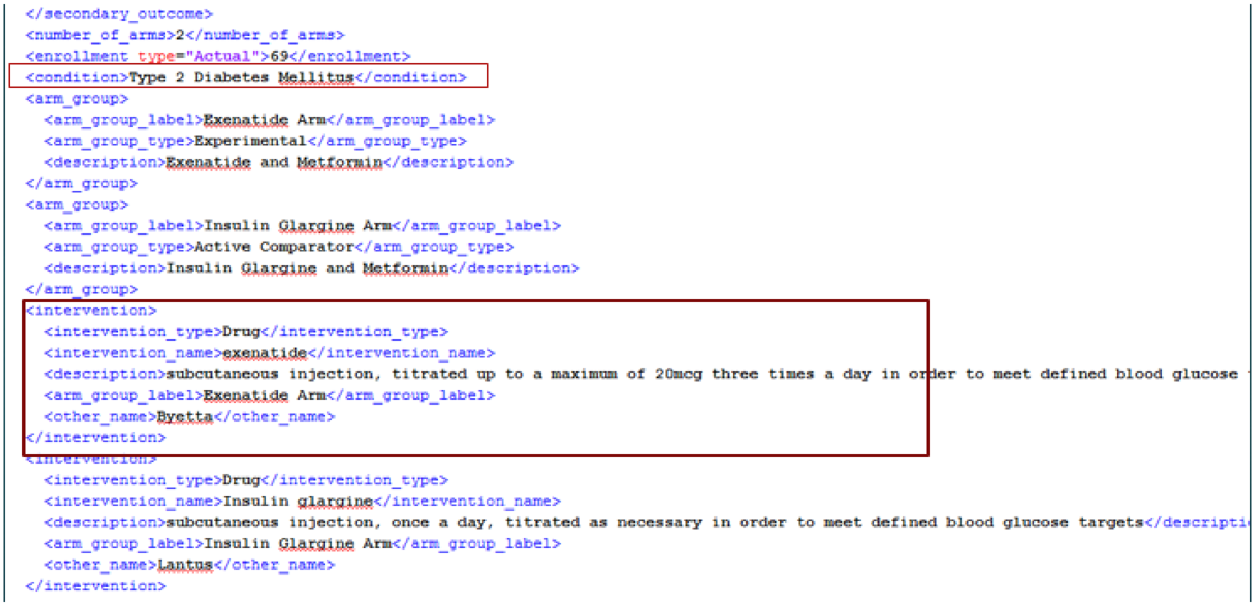
Figure 1: System Architecture

3.3 Clinical Trials Data Extraction and Loading

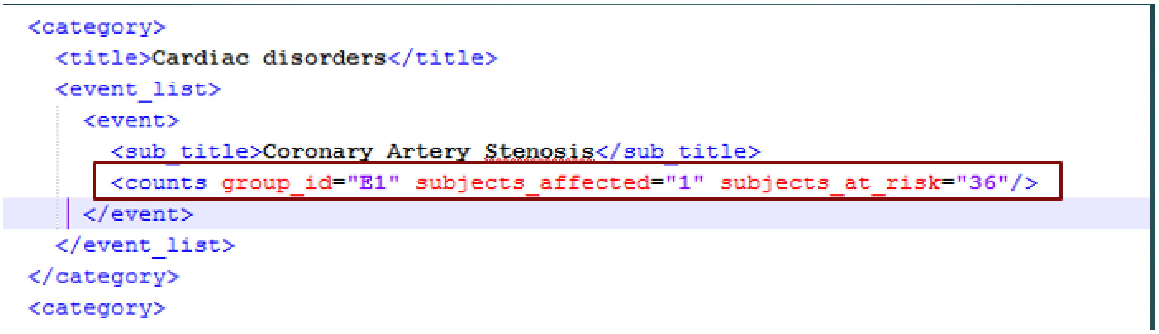
In this section, we will look into the detailed process of extraction and loading of the data. The data is collected from the Clinical Trials [4] in the XML format. Only the studies with results were selected as to get the evidence to query the system depending upon the search terms.

The data downloaded from the clinical trials [4] consists of all the information related to the study like condition, drugs used, side effects, details of the test group used, percentage of people affected by side effects in the test group, duration of the test etc.

The data collected from the clinical trials is loaded into the Hadoop Environment [13] in order to process the data in parallel. This environment provides the platform which can be used for data analysis on large amounts of data in a pretty efficient way by processing the data in parallel. A Hadoop cluster [13] is used, which consists of three data nodes for the processing of the data. The initial task is to load the data into the Hadoop Environment. The loaded data is broken into file splits and stored in the three data nodes with a replication factor of two (i.e. for each split size of 64MB, two copies of the split) are stored in the three data nodes. The namenode of the cluster stores the meta information of the file splits like location and offset of the split. This split enables the parallel access of the data across the cluster.



```
</secondary_outcome>
<number_of_arms>2</number_of_arms>
<enrollment type="Actual">69</enrollment>
<condition>Type 2 Diabetes Mellitus</condition>
<arm_group>
  <arm_group_label>Exenatide Arm</arm_group_label>
  <arm_group_type>Experimental</arm_group_type>
  <description>Exenatide and Metformin</description>
</arm_group>
<arm_group>
  <arm_group_label>Insulin Glargine Arm</arm_group_label>
  <arm_group_type>Active Comparator</arm_group_type>
  <description>Insulin Glargine and Metformin</description>
</arm_group>
<intervention>
  <intervention_type>Drug</intervention_type>
  <intervention_name>exenatide</intervention_name>
  <description>subcutaneous injection, titrated up to a maximum of 20mcg three times a day in order to meet defined blood glucose
  <arm_group_label>Exenatide Arm</arm_group_label>
  <other_name>Byetta</other_name>
</intervention>
<intervention>
  <intervention_type>Drug</intervention_type>
  <intervention_name>Insulin glargine</intervention_name>
  <description>subcutaneous injection, once a day, titrated as necessary in order to meet defined blood glucose targets</descripti
  <arm_group_label>Insulin Glargine Arm</arm_group_label>
  <other_name>Lantus</other_name>
</intervention>
```



```
<category>
  <title>Cardiac disorders</title>
  <event_list>
    <event>
      <sub_title>Coronary Artery Stenosis</sub_title>
      <counts group_id="E1" subjects_affected="1" subjects_at_risk="36"/>
    </event>
  </event_list>
</category>
<category>
```

Figure 2: Clinical Trials XML Data

3.4 Data Filtering and Linking with MetaMap

The data loaded into the Hadoop Environment [13] consists of all the information related to the research studies. We do not need all the information related to the study for building the Evidence based Data Model. We only need the condition, drugs used, side effects observed, percentage of people affected in the group used. For collecting only the required data from the data set we use the MapReduce Paradigm to filter the data.

The MapReduce takes the file split stored in the data node and reads each XML node in the split and checks if it matches with the required nodes and if it matches, then it writes the node name and the node data to the context along with the file split name. All this is done in parallel by the map jobs and once all the map jobs are done, the data written to the context is sent to the reduce job. The reduce job writes the data collected from each map into an XML format with only the required data.

Consider the following example:

Condition - > Prostate Cancer

Drug -> Degarelix

SideEffect -> Vascular disorder

Study ID-> NCT00000300

Group size-> 24

People Affected-> 3

```
MapReduce
Input Clinical Trials data into Map
Data is shared among the maps.
Each document sent to the Map is read and required fields are written to the Context
All the data is aggregated and sent to the Reduce
While Context has data
    Each row is aggregated and written to the External XML file
End While
```

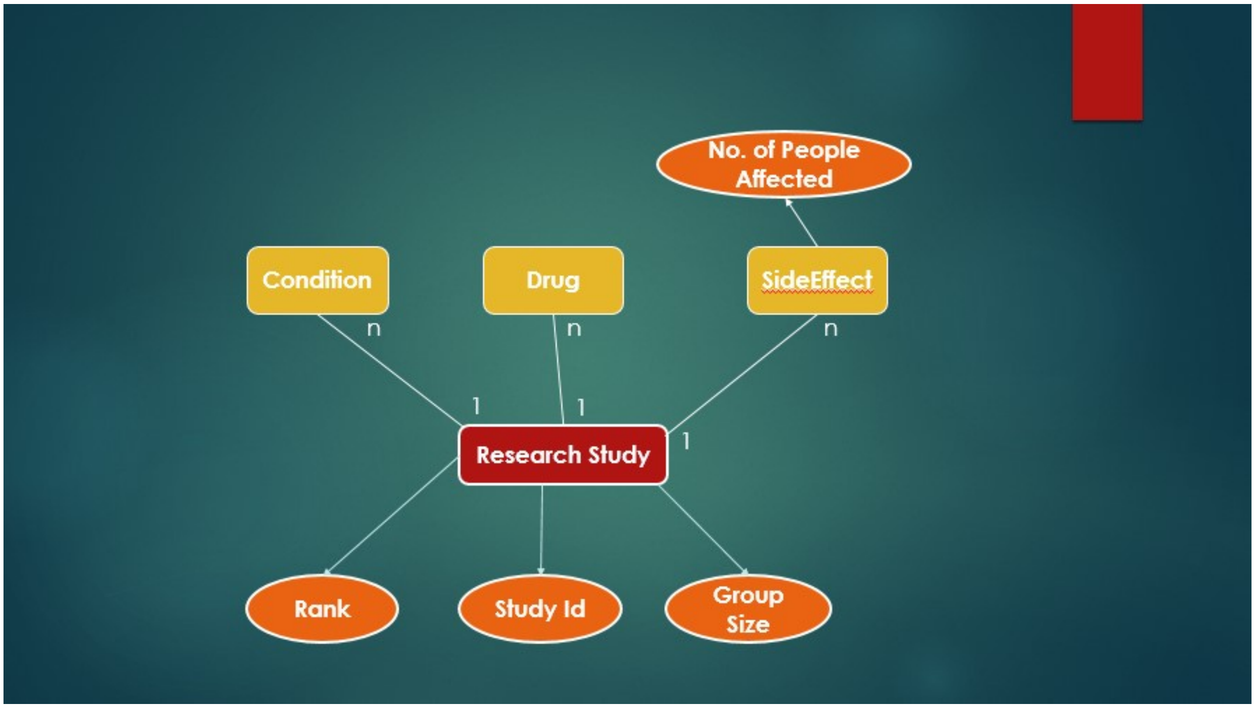


Figure 3: Clinical Trials Data

Once the data is extracted, the conditions are linked with MetaMap [8] to find the related medical terminology of the conditions and find the Meta Score of the condition which relates the condition to the Medical relativeness. This metascore is used to filter the data more from the new XML data set.

Consider the following example:

When "Breast Cancer" term sent as an input to MetaMap we get the following Medical terms : Ductal Carcinoma

Lobular Caricoma

3.5 Building Data Model based on Evidence

The data collected after filtering and linking with MetaMap [8] is used as the data model for the query system. The data is loaded into the MySQL database [18]. Each combination from the XML document is taken along with the related medical terminology and loaded into the table. While loading into the table the frequency of the combination is checked in the table and average of the percentage effected were calculated and stored in the table.

Once the entire data is loaded into the database, the strength of each combination is calculated in the MySQL database [18]. The strength of each combination is calculated by taking the frequency and percentage effected of the combination.

$$\text{Strength} = (\text{Frequency}) / (\text{Percentage affected})^2$$

Frequency = No. of Occurrences of combination in the entire dataset

Percentage affected = People affected percentage among all the test subjects

This formula is in reference to the Medley Clement's formula [10] for comparing groups with respect to the factors.

$$\text{Omega} = (f1)/ (f2)^2$$

Consider the following example:

Prostate Cancer -> Degarelix -> Vascular Disorder -> 7%

Prostate Cancer -> Degarelix -> Vascular Disorder -> 9%

Average Percentage = 8%, Frequency = 2

Strength = $2/(8)^2$ = 0.0312

In Figure 4, the relationship between condition, drug and sideeffect are expalined. Each entity is related to the other using the frequency of occurance of the combination in the entire dataset.

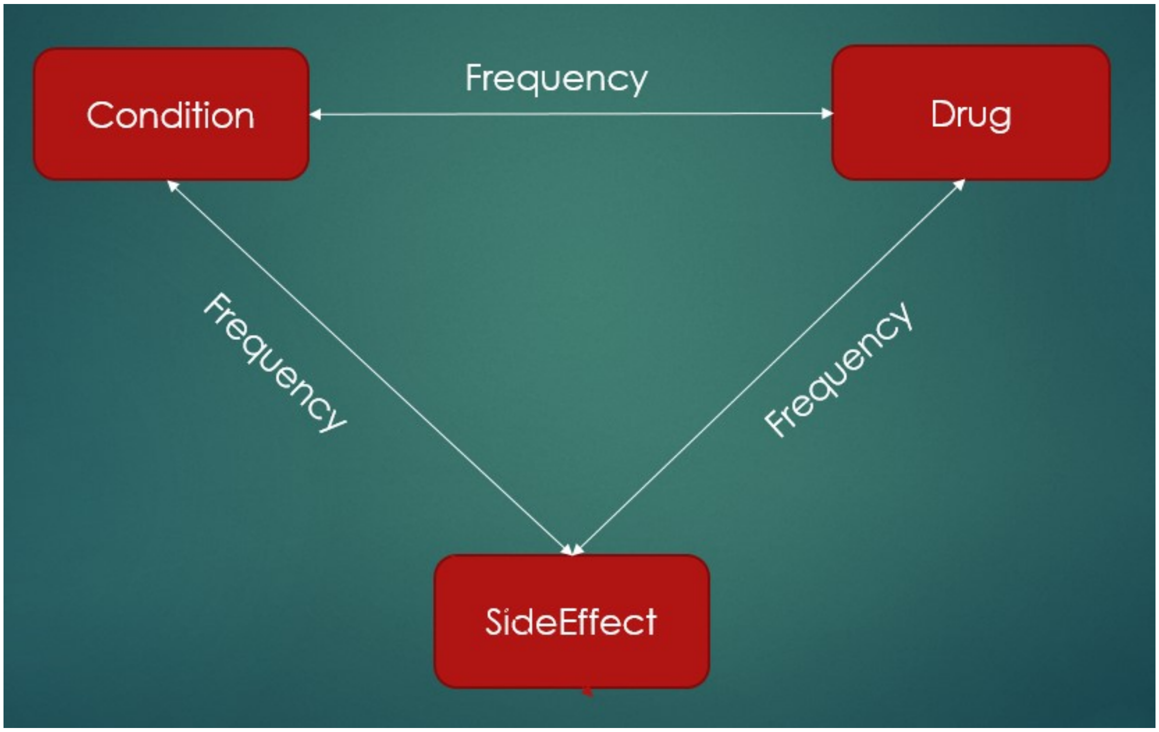


Figure 4: Combination Relationship

In Figure 5, the steps involved in loading the data into the MySQL database are explained.

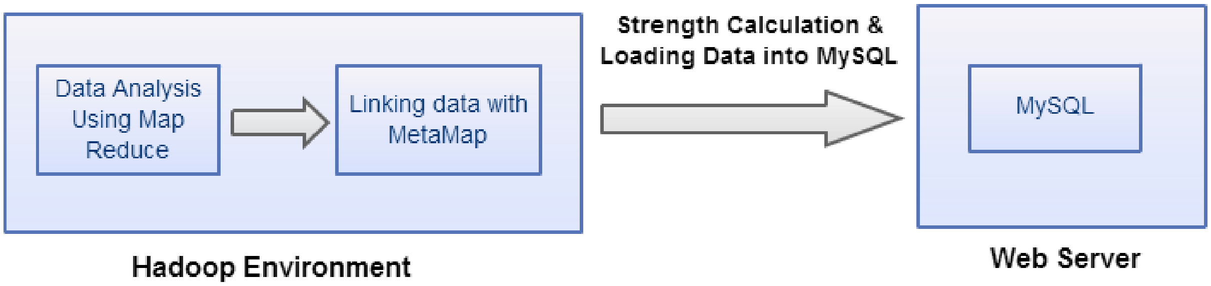

Data Analysis Using Map Reduce
Linking data with MetaMap
Hadoop Environment
Strength Calculation & Loading Data into MySQL
MySQL
Web Server

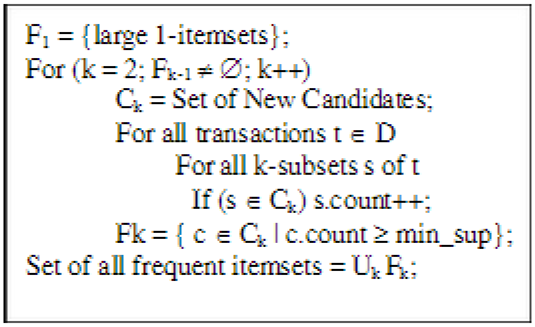
Figure 5: Hadoop to Webserver Migration

3.6 Association Rule Mining

In this section, we will discuss how we used association rule mining on the clinical trials data. The Apriori algorithm is used against the data to find the association rules among the data.

Apriori is the best known association rule mining algorithm. It starts with identifying the frequent individual items in the set and extends them to large item sets.

Apriori Algorithm:



```
$F_1$ = {large 1-itemsets};
For (k = 2; $F_{k-1} \neq \emptyset$; k++)
        $C_k$ = Set of New Candidates;
        For all transactions t $\in$ D
                For all k-subsets s of t
                  If (s $\in$ $C_k$) s.count++;
        Fk = { c $\in$ $C_k$ | c.count $\geq$ min_sup};
Set of all frequent itemsets = $\cup_k F_k$;
```

Figure 6: Apriori Algorithm

Support and Confidence are the two important concepts of Apriori algorithm.

- Support of an itemset Supp(X) is defined as proportion of transactions in the dataset which contains the itemset.

  Ex: {Prostate Cance } => {Blood disorder} has support 20%

  It means 20% of the records in the entire dataset consists of Prostate Cancer and Blood disorder.

- Confidence of a rule is defined as Supp(X U Y)/Supp(X).

  Ex: {Prostate Cancer, Degarelix} => {Blood disorder} has confidence 1

It means 100% records consisting Prostate Cancer, Degarelix also consists of Blood disorder

3.7 Querying the Data Model

In this section, we will discuss how we query the data model. We have built a data model where the combination (Condition, Drug, Side effect) consists of frequency, percentage affected and the strength. Using the strength parameter we present the results to the user which are more effective rather than querying only using keyword search.

When the user queries the system, a SQL query is built using the search term the user has entered and the search type the user selected. The search type consists of three options namely condition, drug and side effect where the user can select the type for querying. The resultant SQL query would look something like this:

```
Select * from table where “search_type” like “%search_query%” order by strength desc
```

The above query searches the table depending upon the search term and order the results by the strength. Because of the ordering of the results using strength, the most effective combination available in the data set will be presented to the user.

3.8 Linking Data Model with PubMed & RxNorm

The data model is linked with PubMed [22] to pull the publications related to the combinations found by querying the model. When the user queries the system, the combinations with the most effective value are returned. PubMed [22] provides an API in order to get publications related to a term. We use this feature of PubMed and pull the publications using all the combinations. In this way, after querying the model for the effective combination, the combination is sent to the PubMed API and this would result in publications consists of all the combinations (Conditions, Drug and Side Effect). Linking of the model with PubMed would provide a platform to obtain more effective publications related to the search term.

The data model is also linked with RxNorm [24], a drug database. RxNorm provides an API to query their data model and find related drug information. The information consists of all the available forms of the drug in market, possible physiological effects, psychological effects, compatibility with other drugs and usage restrictions at the time of particular medical conditions. This information is presented when the user queries the system. Using the drug from each combination, related information is acquired using the RxNorm API.

3.9 Summary

In this section, we have discussed the system architecture of the evidence based medical query on large scale data. Data Extraction and filtering steps were explained in this section. The various steps in building the query model were explained in this section. Overview of how the model is linked with RxNorm and PubMed models were also discussed in this section.

CHAPTER 4

IMPLEMENTATION

4.1 Introduction

In this chapter, we will discuss the complete implementation of the system starting with the setup of the Hadoop cluster followed by MetaMap setup, MapReduce API for data analysis, MetaMap linking, Loading data into MySQL, Calculating the strength, building the PHP query system, linking the results with RxNorm, linking the results with PubMed API.

4.2 Hadoop Cluster Setup

In this section, we will look into detailed procedure of setting up the Hadoop cluster in the local machine. The following steps are involved in Hadoop Cluster setup:

1. In order to build a cluster of Hadoop nodes in a single system, we need to use a virtual box so that each instance of Hadoop sits on one virtual machine. For this we installed VM Ware Workstation, which is an open source application.
2. We downloaded the CentOS iso file to setup linux environment on each virtual instance. For creating a linux instance, I created a new virtual machine in the workstation and loaded the CentOS iso file into the virtual machine. This step will setup the linux environment on the virtual machine.
3. After the setup of the linux environment on the virtual machine, Java is installed on the machine as it is a prerequisite for Hadoop. Java is installed using the following steps:

i. Download latest JDK from Oracle

ii. Extract and install java on the machine. Command : (rpm –Uvh jdk.rpm)

4. Once java is installed, SSH is configured as Hadoop uses it to manage and access each node in the cluster. SSH can be configured using the following steps:

i. Move to the root folder in the terminal. Command : (cd ~)

ii. Command to generate public and private RSA key pair. Command : (ssh-keygen)

iii. Copy public key to authorized keys so that SSH doesn't need password every time. Command : (cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys)

iv. Changing permission of the authorized keys folder. Command :(chmod 700 ~/.ssh/authorized_keys)

5. Next we have installed the Hadoop by downloading the hadoop-1.2.1 which is a stable version available in apache server. Installation is done following the below steps:

i. Extract the files from the download location. Command : (Tar –xvf Hadoop-1.2.1.tar)

6. Then for Hadoop to recognize the java available on the machine, the bashrc file is edited with the Java home and Hadoop home locations.

i. export JAVA_HOME = /usr/java

ii. export HADOOP_HOME = /usr/local/hadoop

iii. export PATH=$PATH:$HADOOP_HOME/bin:$PATH:$JAVA_HOME/bin

7. Now core-site.XML file in the Hadoop conf folder is edited with the following changes:

```
<property>
 <name>fs.default.name</name> <value>hdfs://localhost:54310</value>
</property>
```

8. Then mapred-site.XML file is edited to configure the job tracker location

```
<property>
```

```
<name>mapred.job.tracker</name>

<value>localhost:54311</value>

</property>
```

9. Then hdfs-site.XML file is edited to configure the replication factor of the system.

```
<property>

<name>dfs.replication</name>

<value>1</value>

</property>
```

10. Once the files are edited, then the namenode is formatted in order to start the cluster.

    Format the namenode. Command : (hadoop namenode –format)

11. Then all the services on the cluster are started by using start-all.sh command

12. This process completes the setup of a single node in the cluster.

13. The same process is used to configure 2 more nodes in the cluster.

14. Once all the nodes are setup with the Hadoop, then each node ip address is update in the hosts file in each node

```
gedit /etc/hosts
```

15. Then all the iptables in all the nodes are stopped, so that one node can communicate with other

    Stop the iptables. Command : (Service iptables stop)

16. In this way, the Hadoop multi node cluster is configured.

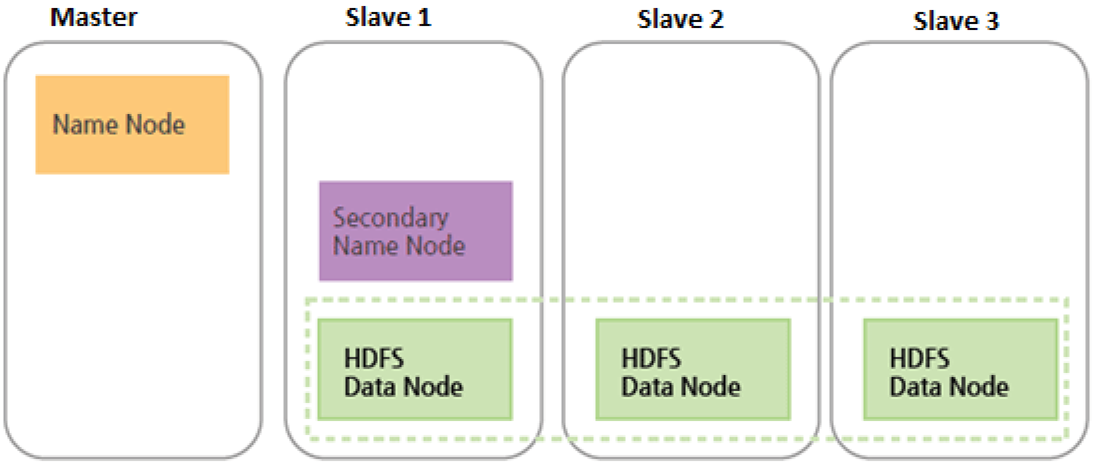
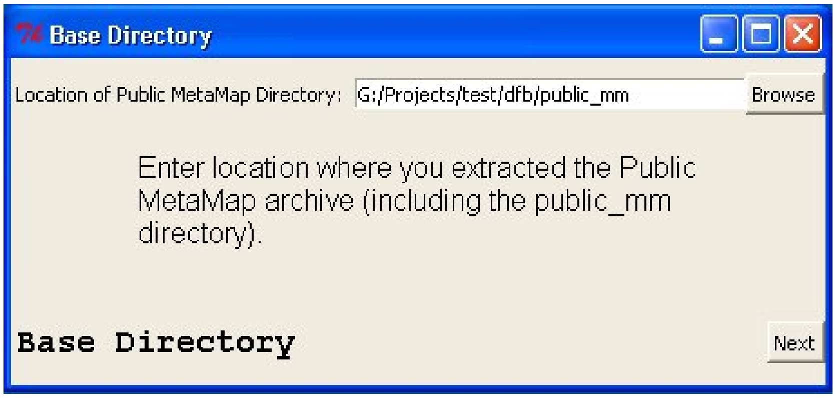


Figure 7: Hadoop Cluster

4.3 MetaMap Setup

In this section, we will look into the detailed procedure of configuring the MetaMap server and the MetaMap Java API in the local machine. The installation procedure is as follows: -

1. The MetaMap distribution 2013 for windows is downloaded from the NLM server [20] by creating an account in the nlm MetaMap server.
2. By running the MetaMap executable file and configuring the distribution location in the machine the installation gets completed.



Location of Public MetaMap Directory: G:/Projects/test/dfb/public_mm

Enter location where you extracted the Public MetaMap archive (including the public_mm directory).

Base Directory

Figure 8: MetaMap Setup

3. Then we started the SKR MEDPOST Part of speech Tagger server to run the MetaMap services.

i. Move to public_mm folder in command prompt

ii. Medpost server is started in order to find the mappings. Command: (bin/skrmedpostctl start.bat)

4. After starting the Medpost server in a new terminal Word Sense Disambiguation (WSD) Server is started

```
bin/wsdserverctl start
```

5. Once the server setup is done, the Java API is installed on the system as the application needs to programmatically communicate with the MetaMap server and pull the required data.

i. Java API folder is extracted into the public installation of the MetaMap files.

ii. Install.exe file is executed in the public directory for configuring the java API.

6. After the installation, the MetaMap server is started.

   Command: (bin/mmserver13)

7. This step wraps up the complete setup of the MetaMap server along with the java API in the local machine.

4.4 MapReduce

In this section, we will discuss how we have used the MapReduce programming paradigm to extract the required data from clinical trials and detailed implementation of the map and the reduce phase.

We have downloaded the entire medical studies with results from the clinical trials website in the XML format. The downloaded data consists of all the information related to the study like the study id, conditions, interventions, age groups, drugs used, description of the drugs, study groups, start date etc. We only require the condition, drugs, side effects if any and the percentage of people affected.

We configured the input and output format of the MapReduce job in the configuration section in the main function. Using a custom Input format class the clinical trials data is loaded into the system. The custom input format is implemented as shown in the Pseudo code mentioned below:

```
XMLInputFormat1 extends TextInputFormat
{       Calls custom XML recorder function();
}
XML recorder{
Takes the input split of the XML file
Start = split.start(); // Sets the start of split as open tag
End = split.end();      // sets the end of split as end tag
Sets the start from the start position
}
```

Input to the Map phase is in the custom format which consists of each node from the XML split read. In the map phase when the input matches to any of the required nodes, then the node name and its value will be written to the context. The map jobs are scattered across all the three nodes and will write the data read to the context in parallel.

```
Map(){
        XMLStreamReader reader; // XML streamreader is created
        //Nodes required in the output are added to string array
   String ele[] = {"condition", intervention_type", "intervention_name", "enrollment", "gender",  };
        //Stream reader is looped till it reaches the end
    while  (reader.hasNext()) {
                //If elements is present in array
                If(ele.contains(reader.next()))
                {
                        // Writes the node name and value to the context for the reduce function
        Context.write(reader.getName(),reader.next());
     }
}
Reader.close(); }
```

Once all the map jobs are finished, then the program moves into the reduce phase. The input to the reduce phase would be in the text format of key value pairs. The reduce phase reads the nodes and their values and writes it to the context in the XML format with only the required fields for the data analysis.

```
Reduce()
{       // For each element in the iterator, the item is written to context
        Foreach(iterator )
        {   // Node name and node value is written to context in XML format
            context.write(new  Text("<property>"), null);
      context.write(node.value, null);
      context.write(new  Text("</property>"), null); } }
```

4.5 MetaMap Linking

In this section, we will discuss about how the medical conditions were linked with MetaMap mappings and all the data that is considered and stored after linking with the MetaMap. The output from the MapReduce is taken and each condition in the input file is taken by using XQuery as mentioned below:

```
expr = xpath.compile("//condition/text()");
```

This query pulls all the conditions from the XML input. A MetaMap object is created and each condition is sent to the object and the preferred names and meta score are pulled and written to a new map file. All the conditions with the meta score greater than 950 were considered.

```
MetaMapAPI API = new MetaMapAPIImpl();

List<Result> resultList = API.processCitationsFromString(terms);

for (Result result : resultList) {

for (Utterance utterance : result.getUtteranceList()) {

            for (PCM pcm : utterance.getPCMList()) {

                  for (Mapping map : pcm.getMappingList()) {

                           for (Ev mapEv : map.getEvList()) {

                              preferred_names.add(condition,mapEv.getPreferredName());

                            if(, mapEv.metaScore() > 950)

                     scores.add(condition, mapEv.metaScore());

                           } } }  }
```

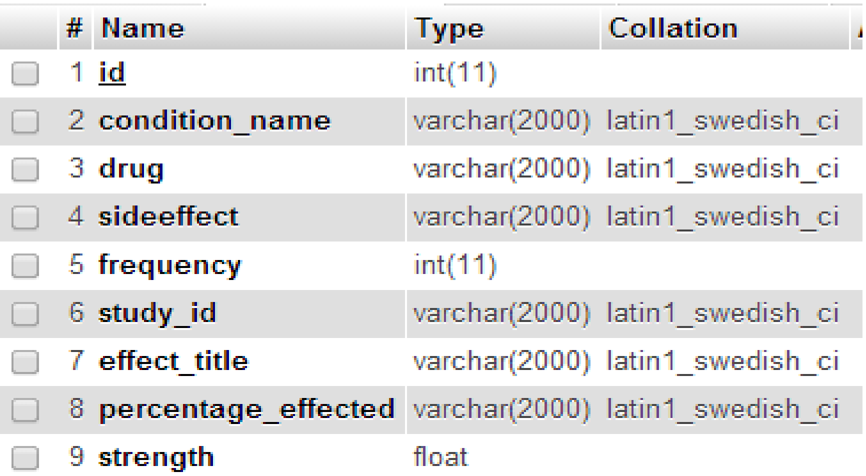
4.6 Data Loading into MySQL

In this section, we will discuss how the data is loaded into the MySQL database and how the frequency and the strength of the Condition, drug and side effect were calculated.

Each study from the output XML of the map reduce job is considered and checked if this is available in the map that is generated by linking to the MetaMap. If this condition is available in the map then the related drug, side effect and the percentage of people effected with this combination are considered. Using a MySQL connector, a connection can be established to the MySQL database from the java program.

```
 Connection con = null;

 String URL = "jdbc:MySQL://localhost:3306/testdb";

 String user = "testuser";

 String password = "test623";

//Establishing Connection to the databse

con = DriverManager.getConnection(URL, user, password);
```

For each combination obtained the Condition, drug and the side effect is checked if available in the table. If available, then the frequency and the percentage effected were taken and the average of the percentage effected is calculated and the frequency value is increased by a value of 1. The structure of the table that stores the below data is as follows:



	#	Name	Type	Collation
□	1	id	int(11)	
□	2	condition_name	varchar(2000)	latin1_swedish_ci
□	3	drug	varchar(2000)	latin1_swedish_ci
□	4	sideeffect	varchar(2000)	latin1_swedish_ci
□	5	frequency	int(11)	
□	6	study_id	varchar(2000)	latin1_swedish_ci
□	7	effect_title	varchar(2000)	latin1_swedish_ci
□	8	percentage_effected	varchar(2000)	latin1_swedish_ci
□	9	strength	float	

Figure 9: Database Table Design

Once all the data is loaded, then the strength of the combination is calculated by taking the frequency and the percentage effected.

$$\text{Strength} = \text{frequency} / (\text{percentage\_effected})^2 \quad \text{// when percentage\_effected != 0}$$

This strength value will be used by the query system to order the results.

4.7 Query System

In this section, we will discuss how the query system is implemented and the technologies used in the implementation process. The data is stored in the MySQL database and SQL is used to retrieve the data from the database. The query page consists of a text box to collect the query text and a dropdown menu to select the type that needed to be queried. The query page looks like this:

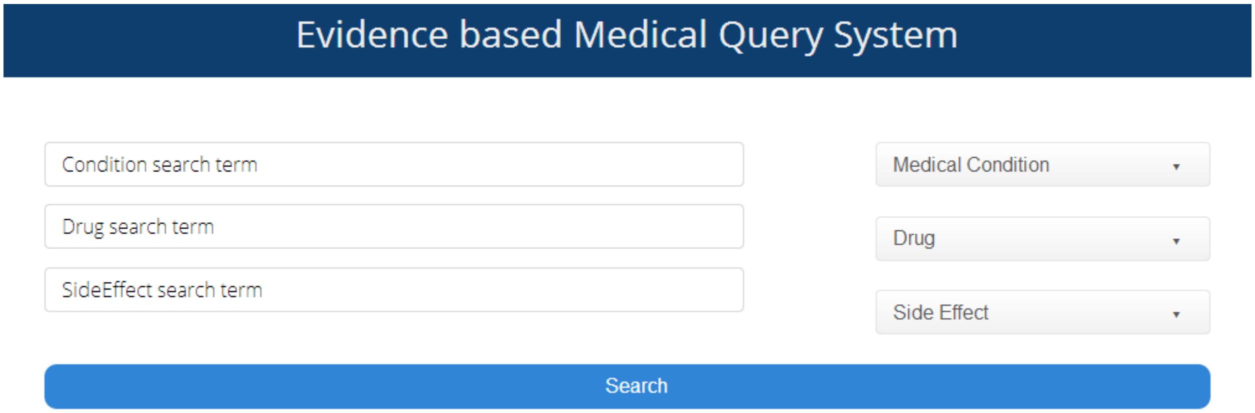
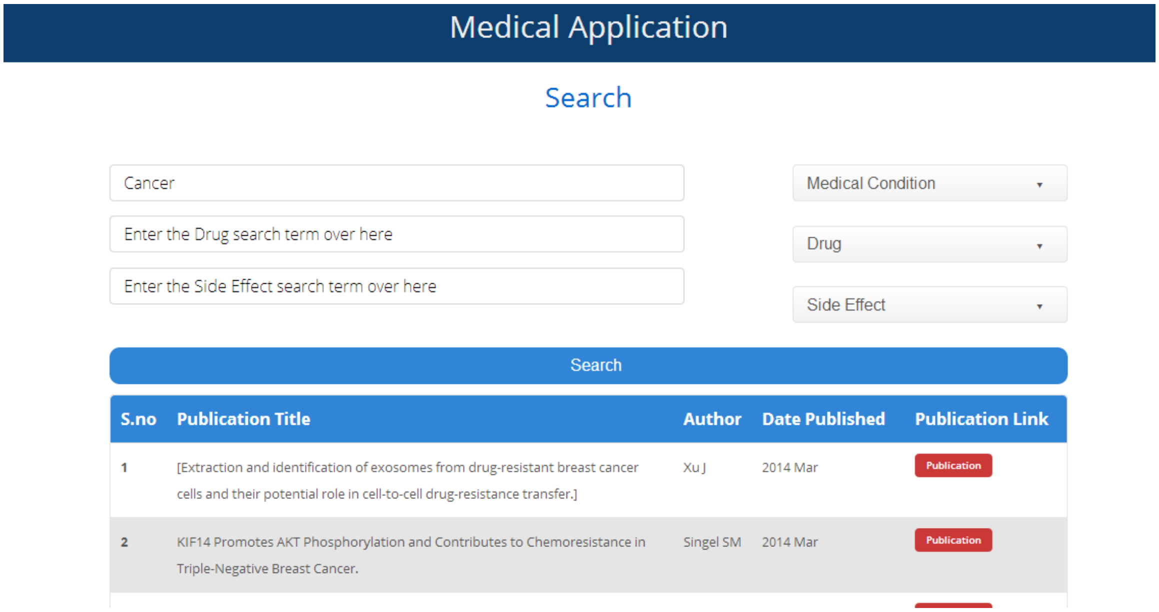


Figure 10: Medical Query System



Cancer

Enter the Drug search term over here

Enter the Side Effect search term over here

S.no	Publication Title	Author	Date Published	Publication Link
1	[Extraction and identification of exosomes from drug-resistant breast cancer cells and their potential role in cell-to-cell drug-resistance transfer.]	Xu J	2014 Mar	Publication
2	KIF14 Promotes AKT Phosphorylation and Contributes to Chemoresistance in Triple-Negative Breast Cancer.	Singel SM	2014 Mar	Publication

Figure 11: Medical Query System PubMed Module

Once a query term is searched, the query term along with the type is sent back to the same page by post method.

```
<form method="POST">

<input type="text" name="query" id="query" value="Enter the search term over here"

        <select name="search_type">

                <option value="condition_name">Medical Condition</option>

                <option value="drug">Drug</option>

                <option value="sideeffect">Side Effect</option>

</select>

<input type="submit" value="Search" /></div>

</form>
```

When the page comes back to the post mode, the post variables i.e. query text and query type are collected and used to run the query on the table. The output results were order by the strength so that the most effective results will be show first.

```
 // When page returned by post request the below code runs

if (strtoupper($_SERVER['REQUEST_METHOD']) == 'POST')

{  // Gets the values from the post opbject

$searchtext = $_POST['query'];

$search_type = $_POST["search_type"];

//  Query to search the table by the requested parameters from the search request

$query = "select condition_name, drug, sideeffect from medical_new where ".$search_type." like '%".$searchtext."%' order by strength desc";

//  Runs the sql query on the database table

$result = MySQLi_query($con, $query);

}
```

The above query returns all the conditions, drug and the side effect that matched the above search text given as input by the user. All the rows returned were looped through and the results were written to the page in the order. The below codes does the writing to the page:

```
// Reads each row returned by the query

while($row = $result->fetch_array())
```

```
{         // Builds the table element in the loop

          echo "<tr>";

                    echo "<td>".$row["condition_name"]."</td>";

                   echo "<td >".$row["drug"]."</td>";

           echo "</tr>";

}
```

4.8 Linking Model with RxNorm

In this section, we will go through all the steps to link with RxNorm API and the information provided by the RxNorm. The Rx Norm API provides the following information for the drugs:

i. Available modes in market
ii. Conditions with which drug cannot be used
iii. Conditions that can be treated with the drug
iv. Physiological Effects
v. Psychological Effects
vi. Drugs that can inhibit the effect of the actual drug
vii. Conditions that can be induced by the drug
viii. Conditions that can be prevented by the drug

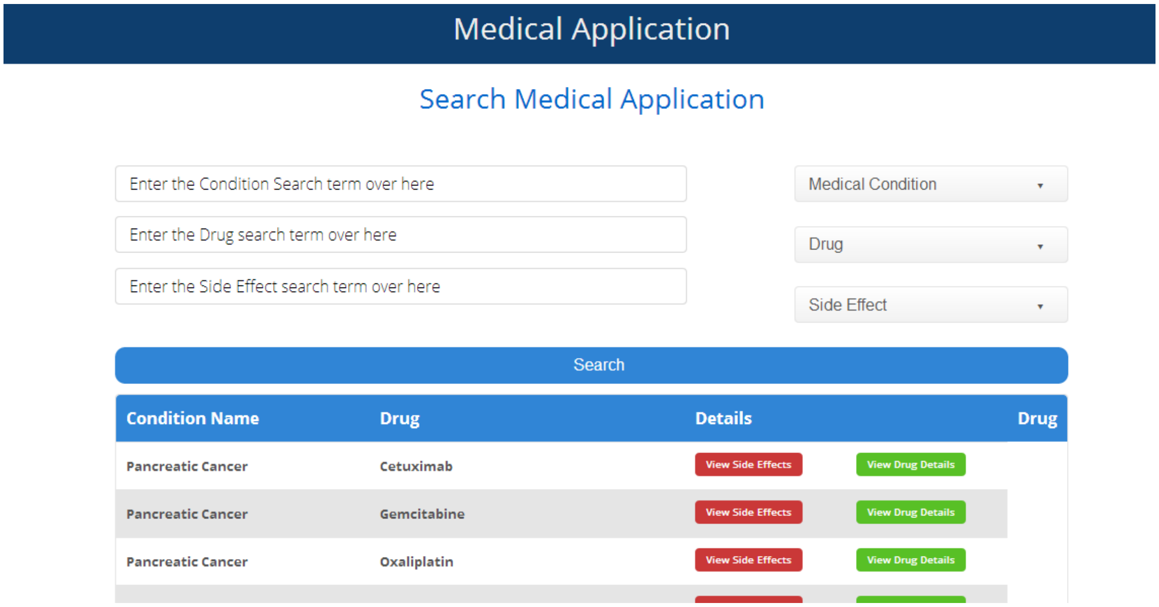
RxNorm provides a RESTful web service. The complete details of the drug can be found by using the Rx Concept id. In order to get the details of the drug, firstly we need to find the concept id and then use that concept id to pull the information. The Concept id can be found by sending the drug name to the following API by appending to the URL. This would give an XML output and the concept id is taken by parsing the XML document resulted from the API call. The code for the API call and parsing the output are mentioned below.

```
// RxNorm web URL for concept id
$URL =
"http://rxnav.nlm.nih.gov/REST/Ndfrt/search?conceptName=".$conceptName."&kindName=DRUG_KIND";
//Call to get the concept name
$XML=simpleXML_load_file($URL);
//Parsing the document
$conceptNui = $XML->groupConcepts->concept->conceptNui;
```

After getting the concept id, this is used to pull the complete information using the Rx Norm API. The below code uses the concept id and pulls all the information.

```
$URL1 = "http://rxnav.nlm.nih.gov/REST/Ndfrt/allInfo/".$conceptNui;
// Call to get all the details
$XML1 = simpleXML_load_file($URL1);
foreach($XML1->fullConcept->groupRoles->role as $child)
```

```
{                   echo $child->roleName . " - ";

                   echo $child->concept->conceptName;

}
```



Medical Application

Search Medical Application

Enter the Condition Search term over here
Enter the Drug search term over here
Enter the Side Effect search term over here
Medical Condition
Drug
Side Effect
Search

Condition Name	Drug	Details		Drug
Pancreatic Cancer	Cetuximab	View Side Effects	View Drug Details	
Pancreatic Cancer	Gemcitabine	View Side Effects	View Drug Details	
Pancreatic Cancer	Oxaliplatin	View Side Effects	View Drug Details	

Figure 12: RxNorm Query System

4.9 Linking Model with PubMed API

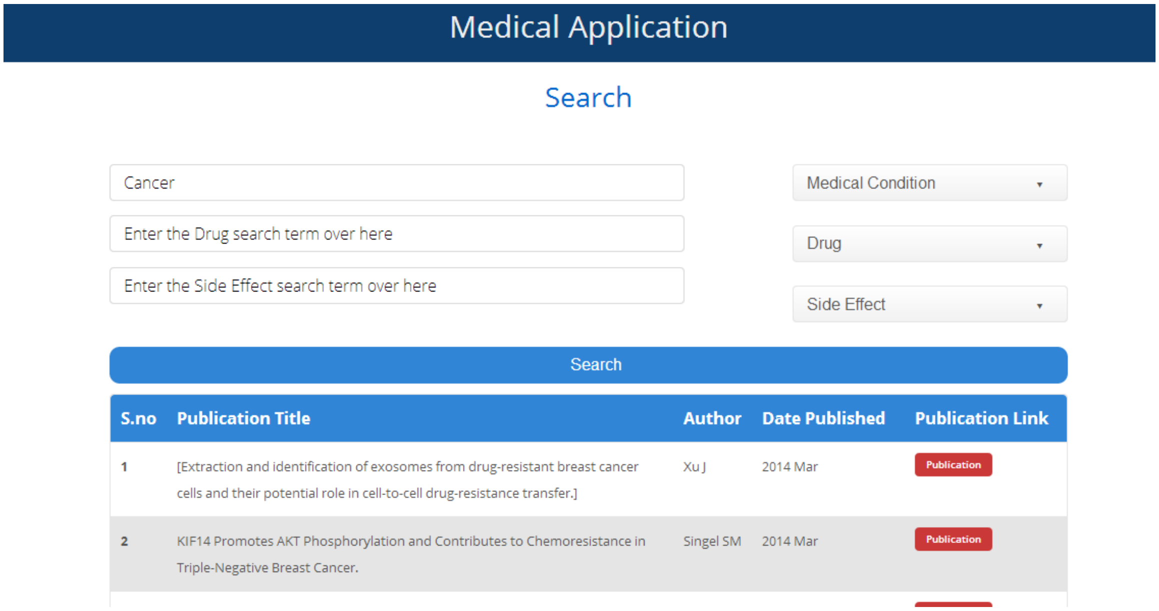
In this section, we will discuss how the query system results were used to pull the publications from the PubMed and how the publications were pulled using the PubMed API.

PubMed provides a RESTful webservice to pull all the publications related to the term passed in the webservice call. PubMed also takes multiple terms input and results the publications that consists of all the terms or the MeSHterms. We took advantage of this feature and built a publication query system on top of the clinical trials data model.

When a user searches for a term using the query type, then the top results with the best strength value are pulled from the database and these terms are sent into the PubMed webservice. This would result all the publications that consists of the terms or MeSH terms. The querying on the system is done similar to the methodology that is mentioned above (Linking with RxNorm). The code to pull the results from the PubMed webservice is as follows:

```
outputString =  “condition_name"+AND + “drug”;

$PubMedAPI = new PubMedAPI();

$idList = $PubMedAPI->getIdString($outputString);

        $XML = $PubMedAPI->getXMLFromIds($idList);

// Gets all the elements from the results returned from the API

    foreach($XML->DocSum as $childDoc)

        {

//  Builds the table to output in the frontend

            $idDetails = $PubMedAPI->getIdDetails($childDoc);

            echo "<tr>";

                echo "<td>".$count."</td>";

                echo "<td>".$idDetails->title."</td>";

                echo "<td>".$idDetails->author."</td>";
```

```
                        echo "<td>".$idDetails->publicationDate."</td>";



                echo "</tr>";


        }
```


Medical Application
Search
Enter the Drug search term over here
Enter the Side Effect search term over here
Medical Condition
Drug
Side Effect
Search

Cancer

S.no	Publication Title	Author	Date Published	Publication Link
1	[Extraction and identification of exosomes from drug-resistant breast cancer cells and their potential role in cell-to-cell drug-resistance transfer.]	Xu J	2014 Mar	Publication
2	KIF14 Promotes AKT Phosphorylation and Contributes to Chemoresistance in Triple-Negative Breast Cancer.	Singel SM	2014 Mar	Publication

Figure 13: PubMed Query System

The entire code for the medical application is uploaded into the Github and the url for the code is: https://github.com/pramodgupta/Evidence-based-Medical-Query-System.

4.10 Summary

In the Implementation section, we have discussed the implementation of each and every step of building the Evidence based Query model. Starting the Hadoop cluster setup, MetaMap setup, linking with MetaMap, the process of loading the data into MySQL, building the query system, linking the model with RxNorm and PubMed were discussed in this implementation section.

CHAPTER 5

RESULTS & EVALUATION

5.1 Overview

In this section, we will discuss about the results that are observed after building the query model. The various medical queries that can be addressed and evaluation of the system were also discussed. The time taken in each process and comparison between centralized vs distributed systems were also mentioned in this section.

5.2 Results

Medical Queries that can be addressed by this application:

All the medical queries that can be addressed by this application, which can be used by the Practitioners were explained in Table 2.

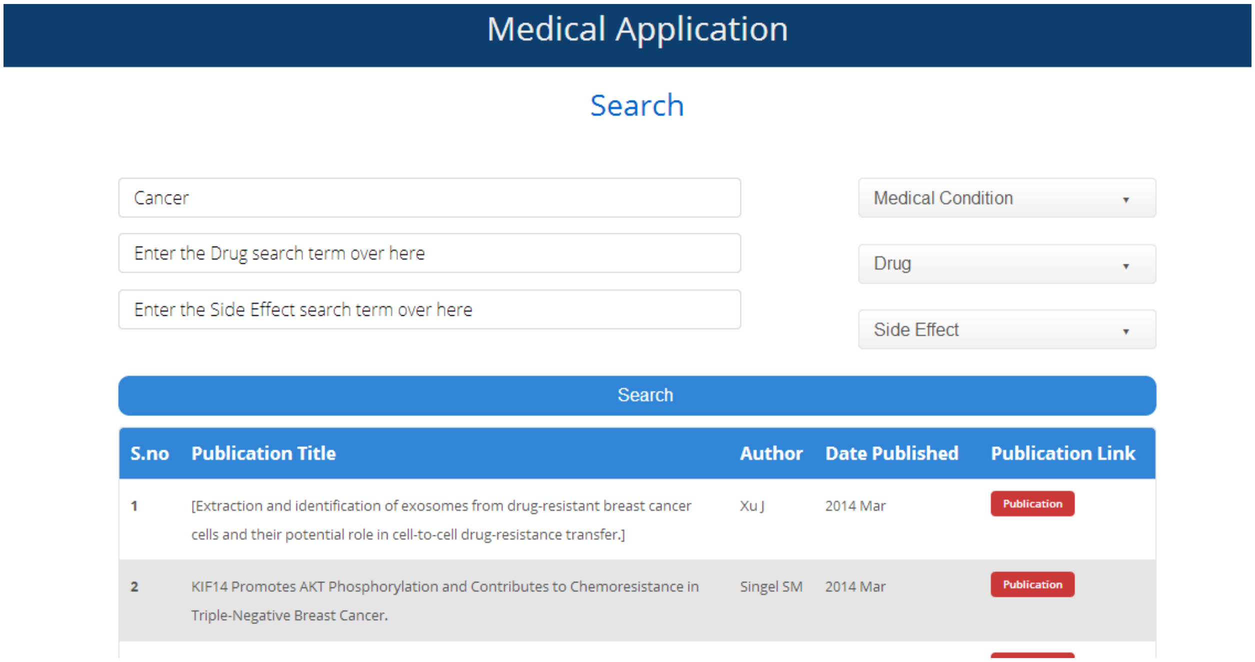
Table 2: Medical Queries

QUERY	RESULT
What is the drug of choice for condition (Prostate Cancer)	Degarelix
What is the cause of symptom (Cardiac Disorder)?	Breast Cancer + docetaxel
What are the sideeffect for condition (Lymphoma)	Respiratory disorder
What drug causes sideeffect (Psychiatric disorders) for condition (Schizophrenia)	Placebo
What sideEffect occurs for Condition(Colorectal Cancer) and drug (Oxaliplatin)	Gastrointestinal disorders

QUERY	RESULT
What drug cannot be used with drug(Docetaxel)	Gemcitabine
Available modes of drug (Docetaxel)	Oxaplatin 100 MG, Oxaplatin 500 MG
All drugs for a Particular Condition	Yes
No. of conditions for which the drug can be used (Degarelix)	3
Percentage of People affected with a particular sideeffect(Vascular Disorder) for a particular drug(Degarelix) and condition(Prostate Cancer) combination	6.25%
Top 5 drugs for Condition (Prostate Cancer)	Prednisone,Carlumab, Abiraterone, Carlumab, Tacrolimus

Search on PubMed Publications based on Medical Condition/ Drug/ Side Effects:

User can search the application by Medical Condition/ Drug/ Side Effect by selecting the option in the drop down and entering the search term in the text box. This would fetch the results from the database with the best combination of drug and side effects and pull the PubMed publications using the PubMed API. Below is the screenshot of the system.


Medical Application
Search
Enter the Drug search term over here
Enter the Side Effect search term over here
Drug
Side Effect
Search

Cancer

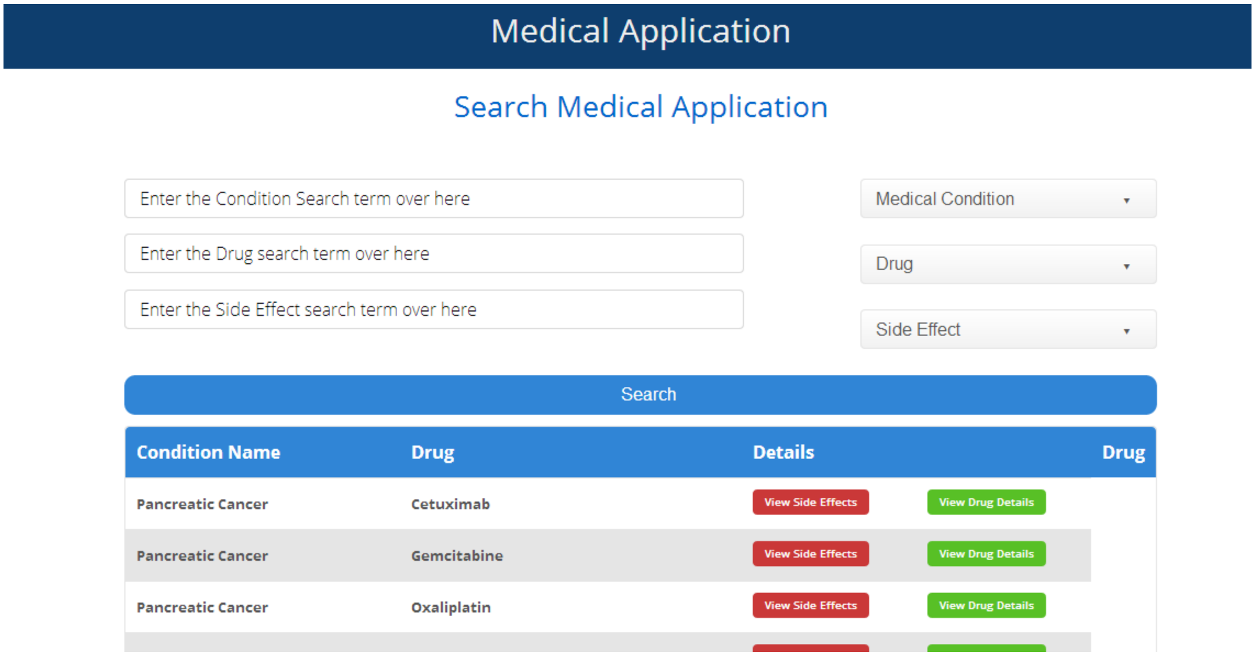
Medical Condition

S.no	Publication Title	Author	Date Published	Publication Link
1	[Extraction and identification of exosomes from drug-resistant breast cancer cells and their potential role in cell-to-cell drug-resistance transfer.]	Xu J	2014 Mar	Publication
2	KIF14 Promotes AKT Phosphorylation and Contributes to Chemoresistance in Triple-Negative Breast Cancer.	Singel SM	2014 Mar	Publication

Figure 14: PubMed Results for Cancer Search Term

Search based on Medical Condition/ Drug/ Side Effect with RxNorm drug details

User can search the application by Medical Condition/ Drug/ Side Effect by selecting the option in the drop down and entering the search term in the text box. This would fetch the results from the database with the best combination of drug and side effects along with the drug details from the RxNorm site using their API. Below is the screenshot of the system.

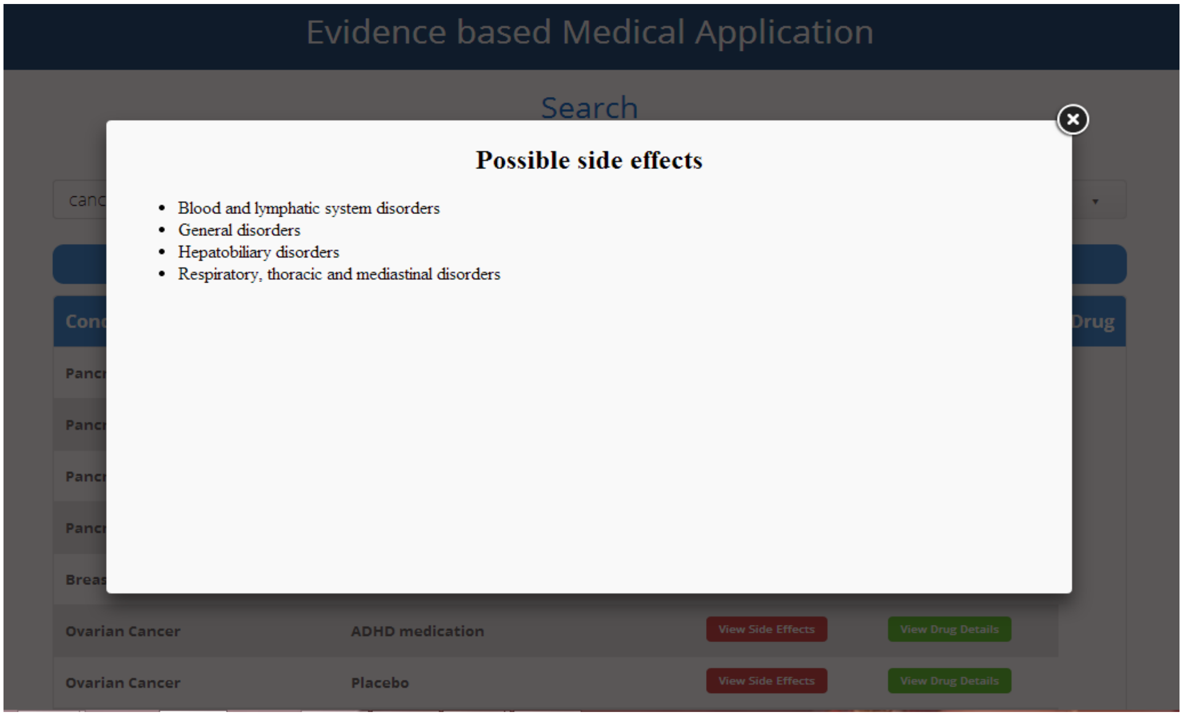


Medical Application
Search Medical Application
Enter the Condition Search term over here
Enter the Drug search term over here
Enter the Side Effect search term over here
Medical Condition
Drug
Side Effect
Search

Condition Name	Drug	Details		Drug
Pancreatic Cancer	Cetuximab	View Side Effects	View Drug Details	
Pancreatic Cancer	Gemcitabine	View Side Effects	View Drug Details	
Pancreatic Cancer	Oxaliplatin	View Side Effects	View Drug Details	

Figure 15: Application search results

The Figure 16, show the side effects for the particular condition and drug combination that were observed in the research studies were mentioned.



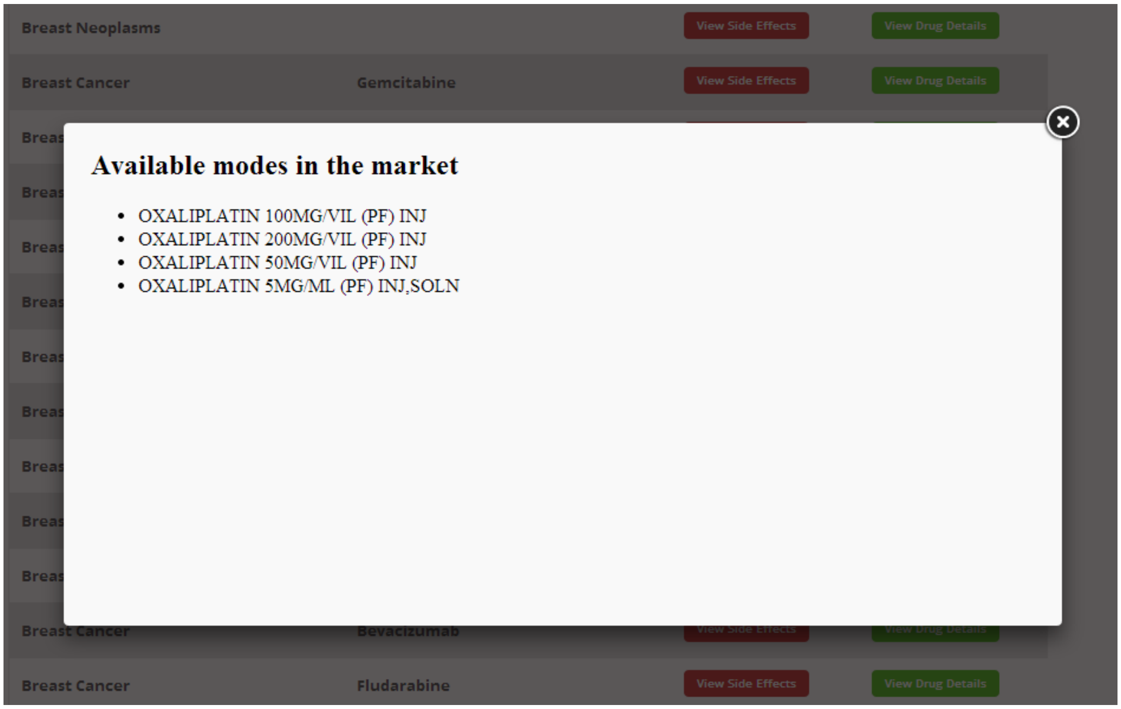
Possible side effects

- Blood and lymphatic system disorders
- General disorders
- Hepatobiliary disorders
- Respiratory, thoracic and mediastinal disorders

Ovarian Cancer	ADHD medication	View Side Effects	View Drug Details
Ovarian Cancer	Placebo	View Side Effects	View Drug Details

Figure 16: Side Effects for Condition & Drug

The drug details like the available versions in the market, the conditions in which drug cannot be used, the conditions that can be treated along with this drug, physiological effects and medical conditions this could induce will be show using the view drug details button.



Available modes in the market

- OXALIPLATIN 100MG/VIL (PF) INJ
- OXALIPLATIN 200MG/VIL (PF) INJ
- OXALIPLATIN 50MG/VIL (PF) INJ
- OXALIPLATIN 5MG/ML (PF) INJ,SOLN

Figure 17: RxNorm Drug Details

5.3 Evaluation

Query comparison between PubMed and Current System

We used the Clinical Trials data along with MetaMap scores and terms as the data source. We ran the same queries on the search application and PubMed and compared the output results.

Table 3: Comparison between PubMed and Application Results

QUERY	APPLICATION RESULTS	PUBMED RESULTS
Medical Condition	All the publications with the condition along with the drug having highest frequency and having less side effect percentage.	All the publications with the medical term ordered by the PubMed criteria.
Drug	All publications with the drug along with the condition and sideeffect having highest frequency and less percentage effected.	All publications with the drug term.
Side Effect	All publications with the side effect along with the condition and drug having highest frequency and less percentage effected.	All publications with the side effect term in the abstract.

Medication Condition (Prostate Cancer)

In the below mentioned table, the PubMed results consists of the publications only with the Prostate Cancer term mentioned , where as the results from my application consists of the all the publications with combination of Prostate Cancer(Medical Condition), Degarelix (Drug) and Nervous disorder(Side Effect).

Table 4: Comparison between PubMed and Application Results for Prostate Cancer

APPLICATION RESULT TITLES	PUBMED RESULT TITLES
Degarelix versus Goserelin plus Bicalutamide Therapy for Lower Urinary Tract Symptom Relief, Prostate Volume Reduction and Quality of Life Improvement in Men with Prostate Cancer. **PubMed ID: 24603064**	*Correlation of Sprouty1 and Jagged1 with Aggressive Prostate Cancer Cells with Different Sensitivities to Androgen Deprivation.* **PubMed ID: 24604720**
A cost-utility analysis of degarelix in the treatment of advanced hormone-dependent prostate cancer. **PubMed ID: 24568188**	*Protocol for the ProCare Trial: a phase II randomised controlled trial of shared care for follow-up of men with prostate cancer.* **PubMed ID: 24604487**
Prostate cancer: Tipping the balance in favour of degarelix for ADT. **PubMed ID: 24473414**	*The likelihood of death from prostate cancer in men with favorable or unfavorable intermediate-risk disease.* **PubMed ID: 24604289**
Disease Control Outcomes from Analysis of Pooled Individual Patient Data from Five Comparative Randomised Clinical Trials of Degarelix Versus Luteinising Hormone-releasing Hormone Agonists. **PubMed ID: 24440304**	*A High-Affinity Near-Infrared Fluorescent Probe to Target Bombesin Receptors.* **PubMed ID: 24604209**
Three-year follow-up of 12 patients with prostate cancer treated with monthly degarelix in a phase II clinical trial. **PubMed ID: 24423954**	*Alpha emitter radium-223 dichloride : New therapy in castration-resistant prostate cancer with symptomatic bone metastases.* **PubMed ID: 24604017**

Drug (Placebo)

In the below mentioned table the results are the output for the search by drug criteria, the PubMed search results consists of all publications with placebo in the abstract, where as the results from the applications consists of all publications with a combination of Depression (Medical Condition), Placebo (Drug) and Psychiatric disorders(Side Effect).

Table 5: Comparison between PubMed and Application Results for Placebo

APPLICATION RESULT TITLES	PUBMED RESULT TITLES
Design and Methodology for the Korean Observational and Escitalopram Treatment Studies of Depression in Acute Coronary Syndrome: K-DEPACS and EsDEPACS. **PubMed ID: 24605129**	*Design and Methodology for the Korean Observational and Escitalopram Treatment Studies of Depression in Acute Coronary Syndrome: K-DEPACS and EsDEPACS.* **PubMed ID: 24605129**
A comparison of responses to the positive and negative syndrome scale (PANSS) between patients with PTSD or schizophrenia. **PubMed ID: 24602497**	*Efficacy and Tolerability of Benzodiazepines for the Treatment of Behavioral and Psychological Symptoms of Dementia: A Systematic Review of Randomized Controlled Trials.* **PubMed ID: 24604893**
Omega-3 Fatty Acids in the Prevention of Interferon-Alpha-Induced Depression: Results from a Randomized, Controlled Trial. **PubMed ID: 24602409**	*Efficacy of combination therapy with erythropoietin and methylprednisolone in clinical recovery of severe relapse in multiple sclerosis.* **PubMed ID: 24604685**

APPLICATION RESULT TITLES	PUBMED RESULT TITLES
Predictors of functional improvement in employed adults with major depressive disorder treated with desvenlafaxine. **PubMed ID: 24583567**	*Efficacy and Safety of Vildagliptin as Add-on to Metformin in Japanese Patients with Type 2 Diabetes Mellitus.* **PubMed ID: 24604395**

Side Effect (Vascular Disorder)

In the below mentioned table the results are the output for the search by side effect criteria, the PubMed search results consists of all publications with vascular disorder in the abstract, whereas the results from the applications consists of all publications with a combination of Colorectal cancer (Medical Condition), Oxalipatin (Drug) and Vascular disorder (Side Effect).

Table 6: Comparison between PubMed and Application Results for Vascular Disorder

APPLICATION RESULT TITLES	PUBMED RESULT TITLES
Se-methylselenocysteine offers selective protection against toxicity and potentiates the antitumour activity of anticancer drugs in preclinical animal models. **PubMed ID: 24619073**	*Decreased tumour necrosis factor alpha (tnf-a) in serum of patients with achilles tendinopathy: further evidence against the role of inflammation in the chronic stage.* **PubMed ID: 24583567**
Improved time to treatment failure with an intermittent oxaliplatin strategy: results of the CONcePT trial. **PubMed ID: 24608198**	*Vascular mediators in chronic lung disease of infancy: Role of endothelial monocyte activating polypeptide II (EMAP II).* **PubMed ID: 24619875**

Five Fractions of Radiation Therapy Followed by 4 Cycles of FOLFOX Chemotherapy as Preoperative Treatment for Rectal Cancer **PubMed ID: 24606849**	*Light chain deposition disease without glomerular proteinuria: a diagnostic challenge for the nephrologist.* **PubMed ID: 24619059**
Chemotherapy for elderly patients with colorectal cancer **PubMed ID: 24739896**	*Extracellular matrix assessment of infected chronic venous leg ulcers: role of metalloproteinases and inflammatory cytokines.* **PubMed ID: 24618232**

CLICNICAL TRIALS DATA ANALYTICS

Top Medical Conditions

All the top conditions observed in the data set were mentioned in Table 7. In the dataset, Breast Cancer is the condition that has been observed in 1536 research studies which makes it around 12% among the entire research studies.

Table 7: Top Ten Medical Conditions

Condition	Frequency
Breast Cancer	1536 (12%)
Prostate Cancer	922 (7%)
Myeloma	809 (6.5%)
Schizophrenia	796 (6.4%)
Leukemia	781 (6.3%)
Lung Cancer	741 (6%)
Ovarian Cancer	729 (5.9%)
Lymphoma	698 (5.6%)

Myelodysplastic Syndromes	609 (4.9%)
Colorectal Cancer	598 (4.8%)

Top Drugs

All the top drugs observed in the research studies were mentioned in the Table 8. Placebo is the drug that has been observed in 3435 studies that makes it around 28% among the entire dataset.

Table 8: Top Ten Drugs

Drug	Frequency
Placebo	3435 (28%)
Cisplatin	1189 (9.6%)
Paclitaxel	1158 (9.3%)
Carboplatin	1118 (9.0%)
Cyclophosphamide	901 (7.3%)
Bevacizumab	886 (7.1%)
Docetaxel	660 (5.3%)
Gemcitabine	610 (4.9%)
Lenalidomide	601 (4.8%)
Pemetrexed	589 (4.7%)

Top Side Effects

All the top side effects observed in the research studies were mentioned in the Table 9. Gastrointestinal disorders is the Side Effect that has been observed in 4451 studies that makes it around 36% among the entire dataset.

Table 9: Top Ten Side Effects

Side Effect	Frequency
Gastrointestinal disorders	4451 (36%)
Nervous system disorders	3856 (31%)
Respiratory, thoracic and mediastina disorders	3781 (30%)
Blood and lymphatic system disorders	3664 (29.7%)
Cardiac disorders	3472 (28%)
Vascular disorders	3147 (25.5%)
Metabolism and nutrition disorders	3101 (25.1%)
Renal and urinary disorders	2750 (22.3%)
Musculoskeletal and connective tissue disorders	2571 (20.8%)
Psychiatric disorders	2386 (19.3%)

Top Condition and Drug Combination

All the top condition and drug combinations observed in the research studies were mentioned in the Table 10. Prostate cancer and Degarelix is the combination that has been observed in 627 studies which makes it around 5% among the entire dataset.

Table 10: Top Ten Condition and Drug Combinations

Condition	Drug	Frequency
Prostate Cancer	Degarelix	627 (5%)
Breast Cancer	Docetaxel	223 (1.8%)
Colorectal Cancer	Oxaliplatin	202 (1.6%)
Myeloma	Dexamethasone	192 (1.5%)
Depressive Disorder	Placebo	162 (1.3%)

Breast Cancer	Cyclophosphamide	161 (1.3%)
Prostatic Neoplasms	Prednisone	158 (1.28%)
Alzheimers Disease	Placebo	154 (1.24%)
Lymphoma	Cyclophosphamide	148 (1.2%)
Leukemia	Cyclophosphamide	139 (1.1%)

Top Drug and Side Effect Combination

All the top Side Effect and Drug combinations observed in the research studies were mentioned in the Table 11. Placebo and Depression disorder is the combination that has been observed in 443 studies which makes it around 3.5% among the entire dataset.

Table 11: Top Ten Drug and Side Effect Combinations

Drug	Side Effect	Frequency
Placebo	Depression disorder	443 (3.5%)
Placebo	Nervous	420 (3.4%)
Placebo	Gastrointestinal disorders	411 (3.3%)
Bevacizumab	Vascular disorder	150 (1.2%)
carboplatin	Blood and lymphatic system disorders	136 (1.1%)
Bevacizumab	Nervous system disorders	134 (1.08%)
Placebo	Vascular disorder	131 (1.06%)
Cisplatin	Gastrointestinal disorders	121 (0.99%)
paclitaxel	Gastrointestinal disorders	119 (0.97%)

cyclophosphamide	Infections and infestations	114 (0.92%)

Top Condition and Side Effect Combination

All the top Condition and Side Effect combinations, observed in the research studies were mentioned in the Table 12. Placebo and Depression disorder is the combination that has been observed in 443 studies which makes it around 3.5% among the entire dataset.

Table 12: Top Ten Condition and Side Effect Combinations

Condition	Side Effect	Frequency
Breast Cancer	Blood and lymphatic system disorders	221 (1.7%)
Breast Cancer	Nervous System disorder	191 (1.54%)
Schizophrenia	Psychiatric disorders	185 (1.5%)
Leukemia	Infections and infestations	151 (1.2%)
Prostate Cancer	Renal and urinary disorders	115 (0.93%)
Prostate Cancer	Gastrointestinal disorders	108 (0.87%)
Lymphoma	Respiratory, thoracic and mediastinal disorders	105 (0.85%)
Breast Cancer	Hepatobiliary disorders	101 (0.81%)
Myeloma	Infections and infestations	93 (0.75%)
Prostate Cancer	Neoplasms benign	91 (0.73%)

Top Condition, Drug and Side Effect Combinations

All the top Condition, Drug and Side Effect combinations, observed in the research studies were mentioned in the Table 13. Prostate Cancer, Nervous system disorder and Degarelix is the combination that has been observed in 443 studies which makes it around 3.5% among the entire dataset.

Table 13: Top Ten Condition, Drug and Side Effect Combinations

Condition	Side Effect	Drug	Strength
Prostate Cancer	Nervous system disorders	Degarelix	1
Prostate Cancer	Gastrointestinal disorders	Degarelix	0.97
Major Depressive Disorder	Psychiatric disorders	Placebo	0.873
Schizophrenia	Psychiatric disorders	Placebo	0.65
Breast Cancer	Cardiac disorders	Docetaxel	0.52
Lymphoma	Respiratory, thoracic and mediastinal disorders	Cyclophosphamide	0.19
Prostate Cancer	Neoplasms benign, malignant and unspecified (incl cysts and polyps)	Degarelix	0.188
Breast Cancer	Blood and lymphatic system disorders	Docetaxel	0.109
Prostate Cancer	Surgical and medical procedures	Degarelix	0.1

Top Apriori Rules

All the top rules that are observed when the Apriori rule mining algorithm is used on the clinical trials data set are mentioned in Table 14.

Table 14: Top Ten Apriori Rules

Condition	Side Effect	Drug	Association
Prostate Cancer	Nervous system disorders	Degarelix	0.94
Prostate Cancer	Gastrointestinal disorders		0.931
Schizophrenia	Psychiatric disorders	Placebo	0.92
Breast Cancer		Docetaxel	0.87
Lymphoma	Respiratory, thoracic and mediastinal disorders	Cyclophosphamide	0.87
Breast Cancer	Blood and lymphatic system disorders	Docetaxel	0.84
Major Depressive Disorder		Placebo	0.71
Prostate Cancer		Degarelix	0.72

Top Common rules from Apriori and Combination Strength method

All the common combinations among the Apriori rule mining method and the Combination strength method are mentioned in Table 15.

Table 15: Common Apriori and Combination Strength Rules

Condition	Side Effect	Drug
Prostate Cancer	Nervous system disorders	Degarelix
Schizophrenia	Psychiatric disorders	Placebo
Breast Cancer	Cardiac disorders	Docetaxel
Lymphoma	Respiratory, thoracic and mediastinal disorders	Cyclophosphamide
Breast Cancer	Blood and lymphatic system disorders	Docetaxel

Combinations vs Association Criteria’s

Figure 18 explains how the no. of combinations changes depending upon the value of combination strength or confidence or support. The graph shows the no. of combinations available at the particular combination strength or confidence or support. When we increase the value of the combination strength or confidence or support the no. of combinations at that particular value decreases.

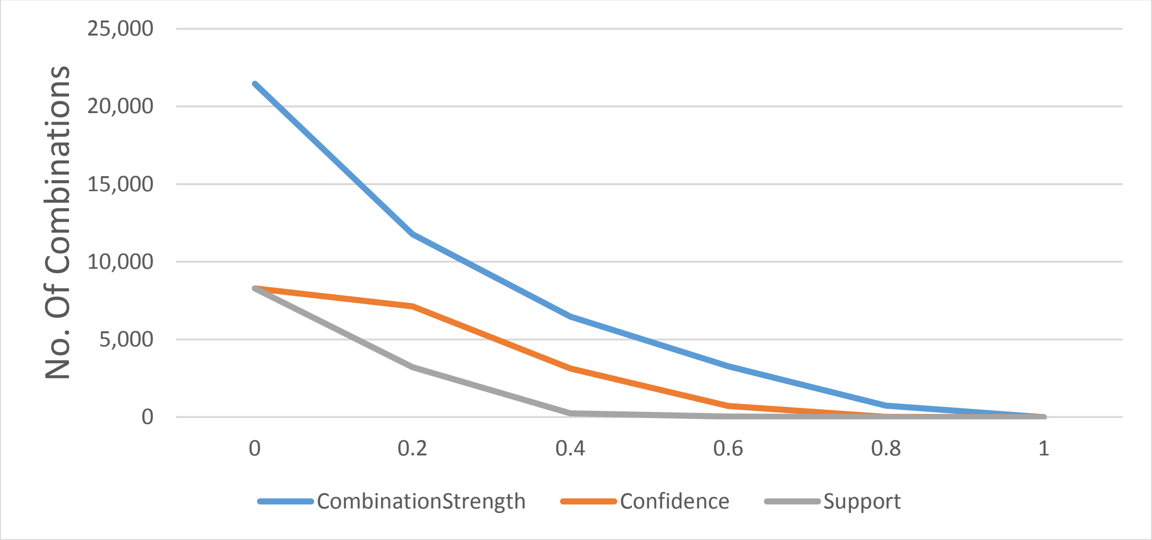


Figure 18: Combinations vs Association Rules

Data Statistics

In Table 16, all the data statistics observed in the clinical trials data set collected were mentioned. Total no. of unique conditions, drugs, side effects observed were also mentioned the in table. Total combinations observed, subjects considered for the research studies and the apriori rules generated were also mentioned in the table.

Table 16: Data Statistics

Type	Count
Conditions	1761
Drugs	2836
Side Effects	27
Total Subjects	432,841
Total research studies	12,327
Total Combinations (Condition / Drug/ Side Effect)	59,228
Total Apriori rules	8,291
Total MetaMap Mappings	9,719
Total Common rules	6,724

Time Comparison

The below chart shows the time comparison between the Distributed system vs Centralized system for the Data Analysis and the MetaMap Linking

Data Analysis: This is the process of analyzing the clinical trials data and write the required data to an external file.

MetaMap Linking: This is the process to pull the Meta score and the related medical terminology of the medical conditions and drugs which can be further used at the time of querying the data model.

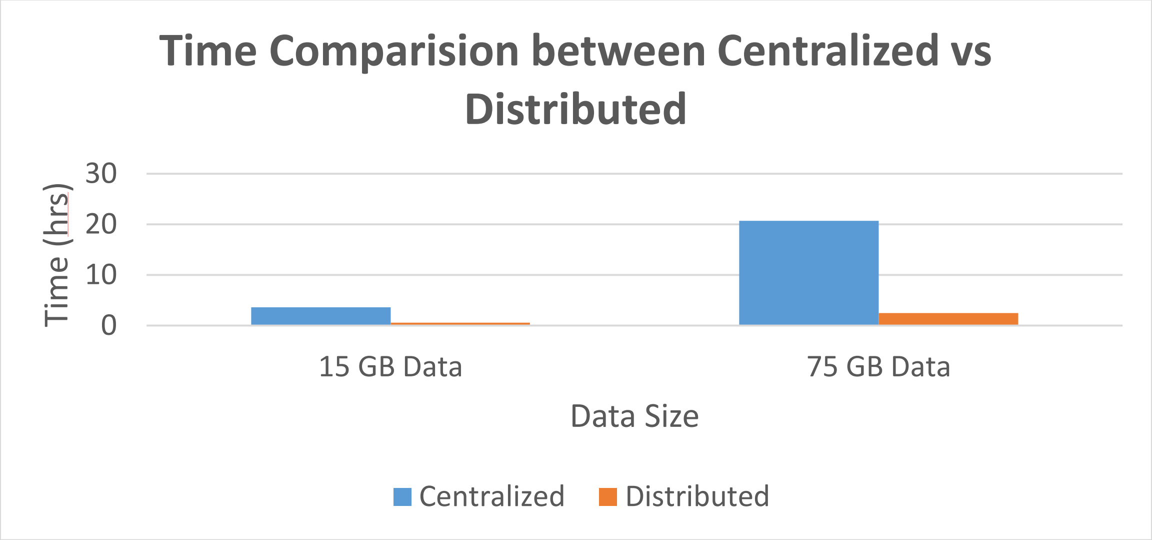


Figure 19: Time Comparison between Centralized vs Distributed

Time taken in Process

The time taken in each process in building the data model is mentioned in the below table.

Table 17: Time in Process

STEP	TIME TAKEN
Loading Clinical trials data into Hadoop	48.7 min
Data Analysis	37.83 min
MetaMap Linking	4.4 min
Association Rule Mining	1.2 min
Export results to MySQL Database	2 min
Query	1.2 sec

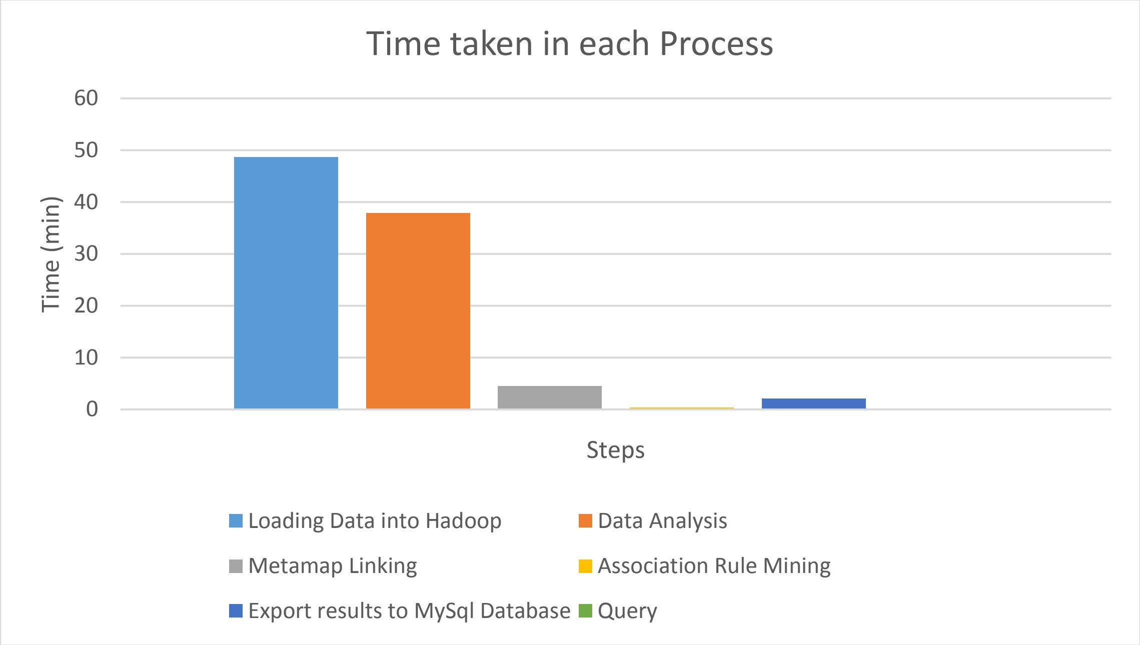


Figure 20: Time taken in Process

5.4 Summary

We have found the various unique drugs, conditions and side effects used in the research studies. We found the best combinations using the rule mining algorithms and combination strength values in the entire research studies. We were able to compare the performance of centralized vs the distributed systems.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this section, the contribution of this thesis is summarized. We presented a novel approach to query the medical data on large scale. We successfully implemented the data analysis of clinical trials data on distributed environment (Hadoop Framework) and link the results with MetaMap to find the meta score and related medical terminology. We calculated the Effective strength of the combination of Condition, Drug and Side effect for querying the system. We found the association rules on the data using the Apriori algorithm.

We have successfully implemented an Evidence based query model using the strength of the combination to present the best combinations possible for the search term. We have successfully integrated this data model with PubMed API and RxNorm to present the publications based on the combinations resulted from querying the system.

6.2 Future Work

In this section, we will discuss about the possible enhancements and extensions of this thesis project. After the data analysis in the distributed system, the output is linked with the MetaMap server in the centralized data nodes. This can be enhanced by implementing the MetaMap linking in distributed environment by hosting the MetaMap server online and linking to the URL directly from the MapReduce.

The results from the MapReduce after linking with the MetaMap is moved to the centralized MySQL database. Instead of moving the data to a centralized system, it can be moved to the HBase tables in distributed system and build the query system on top of HBase in the distributed system.

REFERENCES

[1] Ablimit Aji, Fusheng Wang, Joel Saltz, "Towards building a high performance spatial query system for large scale medical imaging data", Department of Mathematics & Computer Science, Emory University, GA, 2012. Available: http://confluence.cci.emory.edu:8090/download/attachments/6193390/SIGSpatial2012TechReport.pdf?version=1&modificationDate=1350335703000

[2] Apache (1999), Apache Software Foundation, Available: http://www.apache.org/.

[3] Chris Stotle, Diane Tang, and Pat Hanrahan, "A system for query, analysis of multidimensional relational databases", IEEE Transactions on visualization, 2002. Available: http://graphics.stanford.edu/papers/polaris_extended/polaris.pdf

[4] Clinical Trials Data Repository, National Institute of Health, Available: http://www.clinicaltrials.gov/

[5] Cochrane Collaboration (1993), Independent NPO, Available: http://www.cochrane.org/

[6] Creately (2008), Cinergix Inc., Available: https://creately.com

[7] Dennis Wollershiem, "Ontology based query expansion framework for use in medical information systems", La Trobe University, Melbourne, Australia, 2005. Available:http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.59.1543&rep=rep1 &type=pdf

[8] Durk Alan (2005), MetaMap, Available: http://MetaMap.nlm.nih.gov/

[9] Eric Sayers (2008), Entrez Programming Utilities, Available: http://eutils.ncbi.nlm.nih.gov/entrez/eutils/

[10] Gerald Peter Quinin, “Experimental design and data analysis for biologists”, Cambridge University, 2002. Available: http://www.lacbiosafety.org/wp-content/uploads/2011/09/experimental-design-and-data-analysis-for-biologists1.pdf

[11] Giorgio Orsi, Letizia Tanca, “Keyword-based, Context-aware selection of natural language query patterns”, Universitá del Sannio, Italy, 2011. Available: http://www.edbt.org/Proceedings/2011-Uppsala/papers/edbt/a18-orsi.pdf

[12] Google Code (2005), Google Inc. Available : http://code.google.com

[13] Hadoop (2004) , Apache Software Foundation, Available: http://hadoop.apache.org/

[14] HIVE, Apache Software Foundation, Available: http://hive.apache.org/

[15] Jeremy Ginsberg, Matthew H. Mohebbi, “Detecting influenza epidemics using search engine query data”, Nature Vol 457, Google Inc, 2009. Available: http://static.googleusercontent.com/media/research.google.com/en/us/archive/papers/detecting-influenza-epidemics.pdf

[16] Lipyeow Lim, Min Wang, “Supporting keyword based search on medical repositories”, Ph.D, Watson Research Center, Hawthorne, NY, 2008, Available: http://www2.hawaii.edu/~lipyeow/pub/amia08-ontkeyword.pdf

[17] Medline Plus(1998) , NLM, Available: http://www.nlm.nih.gov/medlineplus/

[18] MySQL(1995), Oracle Corporation, Available: http://www.MySQL.com/

[19] National Institute of Health, US Department of Health & Human Sciences, Available: http://www.nih.gov/

[20] NLM API Service (2001), National Library of Medicine, Available: http://www.nlm.nih.gov/API/

[21] Ping Chen, Rakesh Verma, “A Query-based medical information summarization system using ontology knowledge”, University of Houston Downtown, TX,2006, Available:http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1647543&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F10953%2F34552%2F01647543

[22] Pubmed , National Institute of Health, Available: http://www.ncbi.nlm.nih.gov/pubmed

[23] Radu Calinescu, Steve Harris, “Cross-Trial query system for cancer clinical trials”, Computing Laboratory, University of Oxford, , Oxford, UK, 2007, Available: http://www.cs.ox.ac.uk/jeremy.gibbons/publications/crosstrial.pdf

[24] RxNorm, Unified Medical Language System (UMLS), Available: https://www.nlm.nih.gov/research/umls/rxnorm/

[25] RxNorm API (2010), NLM , Available: http://rxnav.nlm.nih.gov/REST/

[26] Stack Overflow (2008), Stack Exchange Inc, Available: http://stackoverflow.com/

[27] Tom Preston (2008), Git Hub , Available: http://github.com

[28] Unified Medical Language System (1986), NLM, Available: http://www.nlm.nih.gov/research/umls/

[29] VMWare Workstation (1999) , VMWare, Available: http://www.vmware.com/products/workstation

[30] Zafar Hashmi , Tatjana Zrimec, " Automatic query generation from computerized clinical guidelines”, International Journal of Information Studies, Vol 1, No 4

(2009),Available:http://www.istudies.net/ojs/index.php/journal/article/viewFile/

61/50

VITA

Venkata Pramod Gupta Bavirisetty was born on March 24, 1990, in Visakhapatnam, India. He completed his schooling in Nellore and graduated high school in 2007. He then completed his Bachelor’s degree in Biotechnology from National Institute of Technology, Warangal, India in 2011. Upon the completion of his Bachelor’s he was placed in Automatic Data Processing as a Software Engineer.

In August 2012, Mr. Venkata Pramod Gupta Bavirisetty came to United States to study Computer Science at the University of Missouri- Kansas City (UMKC), specializing in Software Engineering. Upon completion of his requirements for the Master’s Program, Mr. Venkata Pramod Gupta Bavirisetty plans to work for Epic, Madison.