# 1. INTRODUCTION

People care deeply about their health and want to be, now more than ever, in charge of their health and healthcare. Life is more hectic than has ever been, the medicine that is practiced today is an Evidence-Based Medicine (hereafter, EBM) in which medical expertise is not only based on years of practice but on the latest discoveries as well. Tools that can help us manage and better keep track of our health such as Google Health and Microsoft Health Vault are reasons and facts that make people more powerful when it comes to healthcare knowledge and management. The traditional healthcare system is also becoming one that embraces the Internet and the electronic world. Electronic Health Records (here after, EHR) are becoming the standard in the healthcare domain. Researches and studies show that the potential benefits of having an EHR system are:

**Health information recording and clinical data repositories**

Immediate access to patient diagnoses, allergies, and lab test results that enable better and time-efficient medical decisions

**Medication management**

Rapid access to information regarding potential adverse drug reactions, immunizations, supplies, etc;

**Decision support**

The ability to capture and use quality medical data for decisions in the workflow of healthcare; and

**Obtain treatments that are tailored to specific health needs:**

In order to embrace the views that the EHR system has, we need better, faster, and more reliable access to information. In the medical domain, the richest and most used source of information is Medline, a database of extensive life science published articles. All research discoveries come and enter the repository at high rate (Hunter and Cohen [12]), making the process of identifying and disseminating reliable information a very difficult task. The work presented here is focused on two tasks: automatically identifying sentences published in medical abstracts (Medline) as containing or not information about diseases and treatments, and automatically identifying semantic relations that exist between diseases and treatments, as expressed in these texts. The second task is focused on three semantic relations: Cure, Prevent, and Side Effect.

The tasks that are addressed here are the foundation of an information technology framework that identifies and disseminates healthcare information. People want fast access to reliable information and in a manner that is suitable to their habits and workflow. Medical care related information (e.g., published articles, clinical trials, news, etc.) is a source of power for both healthcare providers and laypeople. Studies reveal that people are searching the web and read medical related information in order to be informed about their health

Our objective for this work is to show what Natural Language Processing (NLP) and Machine Learning (ML) techniques—what representation of information and what classification algorithms—are suitable to use for identifying and classifying relevant medical information in short texts

.

# 2. PROJECT DESCRIPTION

## 2.1 PROBLEM DEFINITION

Healthcare industry has become big business. The healthcare industry produces large amounts of health-care data daily that can be used to extract information for predicting disease that can happen to a patient in future while using the treatment history and health data. This hidden information in the healthcare data will be later used for affective decision making for patient's health. Also, this area need improvement by using the informative data in healthcare. Major challenge is how to extract the information from these data because the amount is very large so some data mining and machine learning techniques can be used. Also, the expected outcome and scope of this project is that if disease can be predicted than early treatment can be given to the patients which can reduce the risk of life and save life of patients and cost to get treatment of diseases can be reduced up to some extent by early recognition. For this problem, a probabilistic modeling and deep learning approach will train a Long Short-Term Memory (LSTM) recurrent neural network and two convolutional neural networks for prediction.The rapid adoption of electronic health records has created wealth of new data about patients, which is a goldmine for improving the understanding of human health. The above method is used to predict diseases using patient treatment history and health data

# 3. COMPUTATIONAL ENVIRONMENT

## 3.1 HARDWARE REQIUREMENTS

| Serial No | Hardware | Description |
|---|---|---|
| 1. | Processor | Pentium IV |
| 2. | Hard Disk | 20GB |
| 3. | RAM | 512 MB |
| 4. | Mouse | 2 Button Mouse |
| 5. | Keyboard | 101 Keys Keyboard |

Hardware requirements are the one that tells us the important and necessary hardware support that are required for the development of the software.

## 3.2 SOFTWARE REQUIREMENTS

The following software's are used to execute the project

| Serial No | Description |
|---|---|
| 1. | Language: Java (JDK 1.6) |
| 2. | Operating System: Windows X.P |
| 3. | Java (as front end) |
| 4. | Oracle 10g (as back end) |

## 3.3 SOFTWARE FEATURES

### Java Technology

Java technology is both a programming language and a platform.

**The Java Programming Language**

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple

- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

**Introduction To Java:**

Java was conceived by James Gosling, Patrick Naughton, Chris Warth, Ed Frank and Mike Sheridan at SUN Microsystems Incorporation in the year 1991.It took 18 months to develop the 1st working version. This language was initially called "OAK", but was renamed "JAVA" in 1995, many more contributed to the design and evolution of the language.

**Java Overview:**

Java is a powerful but lean object-oriented programming language. It has generated a lot of excitement because it makes it possible to program for Internet by creating Applets. Programs that can embedded in web page. The context of an applet can be an animation with sound, an interactive game or a ticker tape. With constantly updated stock prices. Applets can be just little decorations to liven up web page, or they can be serious applications like Word processor or Spreadsheet. Java builds on the strength of C++. It has taken the best features of C++ and discarded the more problematic and error prone parts. To this lean core, it has added garbage collection (automatic memory management), multithreading (the capacity for one program to do more than one thing at a time), security capabilities. This result is that Java is simple, elegant, and powerful and easy-to-use.

Java is actually a platform consisting of 3 components:

1. Java Programming Language.
2. Java Library of Classes and Interfaces.
3. Java Virtual Machine

The following sections will say more about these components.

**Java is portable:**

One of the biggest advantages Java offers is that it is portable. An application written in Java will run on all the major platforms. Any computer with a Java-based browser can run the applications or Applets written in the Java-Programming-Language. A programmer no longer has to write one program to run on a Macintosh, another program to run on a Windows-machine still another to run on a UNIX-machine and so on. In other words, with Java developers write their programs only once.

The Virtual Machine is what gives Java is cross platform capabilities. Rather being compiled into machine language, which is different for each OS's and computer architecture, Java code is compiled into Byte codes.

With other languages, the program code is compiled into a language that the computer can understand. The problem is that other computers with different machine instruction set cannot understand that language.

**Java is Object-Oriented:**

The Java programming language is OBJECT-ORIENTED, which makes program design focus on what you are dealing with, rather than on how you are going to do something. This makes it more useful for programming in sophisticated projects, because one can break the things into understandable components. A big benefit is that these components can then be reused.

Object-Oriented Languages use the paradigm of classes. In simplest term, a class includes both the data and the functions to operate on data. You can create an instance of a class, also called an object, which will have all the data members and functionality of its class. Because of this, you can think of a class as being like template, with each object being a specific instance of a particular type of class.

The class paradigm allows one to encapsulate data so that specific data values are those using the data cannot see the function implementation. Encapsulation makes it possible to make the changes in code without breaking other programs that use that code.

If for example, the implementation of a function is changed, the change is invisible to any programmer who invokes that function, and does not affect his/her program, except hopefully to improve it.

Java includes inheritance, or the ability to derive new classes from existing classes. The derived class, is also called as Sub-Class, inherits all the data in the functions of the existing class.

**Multithreading:**

We are probably familiar with multitasking in your operating system: the ability to have more than one program working at what seems like the same time. For example, you can print while editing or sending a fax. Of course, unless you have a multiple-processor machine, the operating system is really doling out CPU time to each program, giving the impression of parallel activity. This resource distribution is possible because although you may think you are keeping the computer busy by, for example, entering data, much of the CPU's time will be idle.

Multitasking can be done in two ways, depending on whether the operating system interrupts programs without consulting with them first or whether programs are only interrupted when they are willing to yield control. The former is called preemptive multitasking; the latter Sis called cooperative (or, simply, non-preemptive) multitasking. Older operating systems such as Windows 3.x and Mac OS 9 are cooperative multitasking systems, as are the operating systems on simple devices such as cell phones. UNIX/Linux, Windows NT/XP (and Windows 9x for 32-bit programs), and OS X are preemptive. Although harder to implement, preemptive multitasking is much more effective. With cooperative multitasking, a badly behaved program can hog everything.

Multithreaded programs extend the idea of multitasking by taking it one level lower: individual programs will appear to do multiple tasks at the same time. Each task is usually called a thread—which is short for thread of control. Programs that can run more than one thread at once are said to be multithreaded.

So, what is the difference between multiple processes and multiple threads? The essential difference is that while each process has a complete set of its own variables, threads share the same data. This sounds somewhat risky. However, shared variables make communication between threads more efficient and easier to program than inter process communication. Moreover, on some operating systems threads are more “lightweight” than processes—it takes less overhead to create and destroy individual threads than it does to launch new processes. Multithreading is extremely useful in practice. For example, a browser should be able to simultaneously download multiple images. A web server needs to be able to serve concurrent requests. The Java programming language itself uses a thread to do garbage collection in the background—thus saving you the trouble of managing memory! Graphical user interface (GUI) programs have a separate thread for gathering user interface events from the host operating environment. This chapter shows you how to add multithreading capability to your Java applications.

**Collections:**

As the name indicates, collections is a group of objects known as its elements. Basically it is a package of data structures includes Array Lists, Linked Lists,Hash Sets, etc. A collection is simply an object that groups multiple elements into a single unit. It is also called as a container sometimes. It is used to store, retrieve, manipulate, and communicate aggregate data. Typically, it represents data items that form a natural group and allows duplicate elements while others do not. It consists of both ordered and unordered elements. There is no direct implementation of this interface however SDK provides implementations of more specific sub interfaces like Set and List. The manipulation and passing of collections is done by this interface.

The Two "standard" constructors should be provided by all the general-purpose Collection implementation classes. These classes typically implement Collection indirectly through one of its sub interfaces. Void (no arguments) constructor which creates an empty collection.

1. Constructor with a single argument of type Collection, which creates a new collection with the same elements as its argument.

The user can copy any collection using void constructor to produce an equivalent collection of the desired implementation type. As interfaces cannot contain constructors there is no way to enforce this convention. However all of the general-purpose Collection implementations comply this in the Java platform libraries.

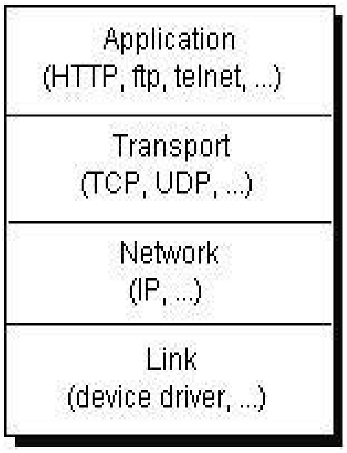
**The Java Collections API:**

Java Collections of API (Application Programming Interface) Consists of several interfaces, and classes that implement those interfaces, within the java.util package. It provides tools for maintaining a data container of objects. Each primitive value must be wrapped in an object of its appropriate wrapper class (Boolean, Character, Integer, Double, etc.) to maintain a collection of primitive data. It is an alternative tool to the creation of custom data structures.

You must be familiar with collections if you have worked with Java programming language. Collection implementations included Vector, Hash table, and array are available in earlier (pre-1.2) versions of the Java platform, but those versions do not include the collections framework. Hence the new version of the Java platform contains the collections framework.

**Networking:**

Computers running on the Internet communicate to each other using either the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP), as this diagram illustrates:



When you write Java programs that communicate over the network, you are programming at the application layer. Typically, you don't need to concern yourself with the TCP and UDP layers. Instead, you can use the classes in the java.net package. These classes provide system-independent network communication. However, to decide which Java classes your programs should use, you do need to understand how TCP and UDP differ.

**The Java Database Connectivity (JDBC)**

JDBC is Java application programming interface that allows the Java programmers to access database management system from Java code. It was developed by

Java Soft, a subsidiary of Sun Microsystems.The Java Database Connectivity (JDBC) API is the industry standard for database-independent connectivity between the Java programming language and a wide range of databases SQL databases and other tabular data sources, such as spreadsheets or flat files. The JDBC API provides a call-level API for SQL- based database access.

JDBC technology allows you to use the Java programming language to exploit "Write Once, Run Anywhere" capabilities for applications that require access to enterprise data. With a JDBC technology-enabled driver, you can connect all corporate data even in a heterogeneous environment.

Java Database Connectivity in short called as JDBC. It is a java API which enables the java programs to execute SQL statements. It is an application programming interface that defines how a java programmer can access thedatabase in tabular format from Java code using a set of standard interfaces and classes written in the Java programming language.

JDBC has been developed under the Java Community Process that allows multiple implementations to exist and be used by the same application. JDBC provides methods for querying and updating the data in Relational Database Management system such as SQL, Oracle etc.

The Java application programming interface provides a mechanism for dynamically loading the correct Java packages and drivers and registering them with the JDBC Driver

Manager that is used as a connection factory for creating JDBC connections which supports creating and executing statements such as SQL INSERT, UPDATE and DELETE. Driver Manager is the backbone of the JDBC architecture.

Generally all Relational Database Management System supports SQL and we all know that Java is platform independent, so JDBC makes it possible to write a single database application that can run on different platforms and interact with different Database Management Systems.

Java Database Connectivity is similar to Open Database Connectivity (ODBC) which is used for accessing and managing database, but the difference is that JDBC is designed specifically for Java programs, whereas ODBC is not depended upon any language.

In short JDBC helps the programmers to write java applications that manage these three programming activities:

1. It helps us to connect to a data source, like a database.

2. It helps us in sending queries and updating statements to the database and

3. Retrieving and processing the results received from the database in terms of answering to your query.

**JDBC Driver Manager**

The JDBC Driver Manager class defines objects which can connect Java applications to a JDBC driver. Driver Manager has traditionally been the backbone of the JDBC architecture. It is quite small and simple.

This is a very important class. Its main purpose is to provide a means of managing the different types of JDBC database driver. On running an application, it is the Driver Manager's responsibility to load all the drivers found in the system property jdbc.drivers. For example, this is where the driver for the Oracle database may be defined. This is not to say that a new driver cannot be explicitly stated in a program at runtime which is not included in jdbc.drivers. When opening a connection to a database it is the Driver Manager’s role to choose the most appropriate driver from the previously loaded drivers.

The JDBC API defines the Java interfaces and classes that programmers use to connect to databases and send queries. A JDBC driver implements these interfaces and classes for a particular DBMS vendor.

A Java program that uses the JDBC API loads the specified driver for a particular DBMS before it actually connects to a database. The JDBC Driver Manager class then sends all JDBC API calls to the loaded driver.

**Connection:**

This is an interface in java.sql package that specifies connection with specific database like: MySQL, Ms-Access, Oracle etc and java files. The SQL statements are executed within the context of the Connection interface.

**Class.forName (String driver):**

This method is static. It attempts to load the class and returns class instance and takes string type value (driver) after that matches class with given string.

**getConnection (String url, String userName, String password):**

This method establishes a connection to specified database url. It takes three string types of arguments like:

url: - Database url where stored or created your

database userName: - User name of MySQL

password: -Password of MySQL

**con.close():**

This method is used for disconnecting the connection. It frees all the resources occupied by the database.

**printStackTrace ():**

The method is used to show error messages. If the connection is not established, then exception is thrown and prints the message.

**Swings:**

The Swing toolkit includes a rich set of components for building GUIs and adding interactivity to Java applications. Swing includes all the components you would expect from a modern toolkit: table controls, list controls, tree controls, buttons, and labels.

Swing is far from a simple component toolkit, however. It includes rich undo support, a highly customizable text package, integrated internationalization and accessibility support. To truly leverage the cross-platform capabilities of the Java platform, Swing supports numerous look and feels, including the ability to create your own look and feel. The ability to create a custom look and feel is made easier with Synth, a look and feel specifically designed to be customized. Swing wouldn't be a component toolkit without the basic user interface primitives such as drag and drop, event handling, customizable painting, and window management.

Swing is part of the Java Foundation Classes (JFC). The JFC also include other features important to a GUI program, such as the ability to add rich graphics functionality and the ability to create a program that can work in different languages and by users with different input devices.The following list shows some of the features that Swing and the Java Foundation Classes provide.

**Swing GUI Components:**

The Swing toolkit includes a rich array of components: from basic components, such as buttons and check boxes, to rich and complex components, such as tables and text. Even deceptively simple components, such as text fields, offer sophisticated functionality, such as formatted text input or password field behavior. There are file browsers and dialogs to suit most needs, and if not, customization is possible. If none of Swing's provided components are exactly what you need, you can leverage the basic Swing component functionality to create your own.

**Java 2D API:**

To make your application stand out; convey information visually; or add figures, images, or animation to your GUI, you'll want to use the Java 2D API. Because Swing is built on the 2D package, it's trivial to make use of 2D within Swing components. Adding images, drop shadows, compositing — it's easy with Java 2D.

**Pluggable Look-and-Feel Support:**

Any program that uses Swing components has a choice of look and feel. The classes shipped by Oracle provide a look and feel that matches that of the platform. The Synth package allows you to create your own look and feel. The GTK+ look and feel makes hundreds of existing look and feels available to Swing programs.A program can specify the look and feel of the platform it is running on, or it can specify to always use the Java look and feel, and without recompiling, it will just work. Or, you can ignore the issue and let the UI manager sort it out.

**Data Transfer:**

Data transfer, via cut, copy, paste, and drag and drop, is essential to almost any application. Support for data transfer is built into Swing and works between Swing components within an application, between Java applications, and between Java and native applications.

**Internationalization:**

This feature allows developers to build applications that can interact with users worldwide in their own languages and cultural conventions. Applications can be created that accept input in languages that use thousands of different characters, such as Japanese, Chinese, or Korean. Swing's layout managers make it easy to honor a particular orientation required by the UI. For example, the UI will appear right to left in a locale where the text flows right to left. This support is automatic: You need only code the UI once and then it will work for left to right and right to left, as well as honor the appropriate size of components that change as you localize the text.

**Accessibility API:**

People with disabilities use special software — assistive technologies — that mediates the user experience for them. Such software needs to obtain a wealth of information about the running application in order to represent it in alternate media: for a screen reader to read the screen with synthetic speech or render it via a Braille display, for a screen magnifier to track the caret and keyboard focus, for on-screen keyboards to present dynamic keyboards of the menu choices and toolbar items and dialog controls, and for voice control systems to know what the user can control with his or her voice. The accessibility API enables these assistive technologies to get the information they need, and to programmatically manipulate the elements that make up the graphical user interface.

**Undo Framework API:**

Swing's undo framework allows developers to provide support for undo and redo. Undo support is built in to Swing's text component. For other components, Swing supports an unlimited number of actions to undo and redo, and is easily adapted to an application. For example, you could easily enable undo to add and remove elements from a table.

**Flexible Deployment Support:**

If you want your program to run within a browser window, you can create it as an applet and run it using Java Plug-in, which supports a variety of browsers, such as Internet Explorer, Firefox, and Safari. If you want to create a program that can be launched from a browser, you can do this with Java Web Start. Of course, your application can also run outside of browser as a standard desktop application.

## Back End

**ORACLE 10G**Oracle is an Object Relational Database Management system (ORDBMS). It offers capabilities of both relational and object oriented database systems.

## New Features of Oracle 10g

The overall focus of the Oracle 10g new features is of course grid computing, the idea of which is the introduction of computing power available on demand much like electricity. Or as Giga Research put it: "In basic terms grids are clusters of computers or servers that are linked together enabling the pooling of computing resources" ("Defining Grid Computing", Giga Research, August 2002).

The main driver for this focus in Oracle 10g is the fact that many if not most organizations have more computing resources than they actually need most of the time, due to systems being sized to handle the peak load which might occur just once per year. In fact research by Forrester concluded "Globally, firms spend more than $49 billion a year on servers and CIOs report that server utilization is 60% or less". (Forrester Research, April 2002)

The specific Oracle 10g new features that take advantage of the grid are Real Application Clusters (not entirely new but enhanced from Oracle 9i), Automatic Storage Management (ASM) new in Oracle 10g, Oracle Streams and Oracle Resource Manager (again, both of these have been enhanced from Oracle 9i). Underneath the grid computing umbrella, the Oracle 10g new features can be organized into 5 broad categories:

1. Manageability/automation - in other words decreasing the amount of time spent on routine tasks to leave more time for strategic tasks

    a. Common Manageability Infrastructure

    b. Storage Enhancements

    c. Space Management

    d. Other Manageability Enhancements

    e. Human Error Correction (including flashback database)

2. Availability

    a. Recovery through reset logs

    b. Backup and recovery

    c. Data Guard

    d. Online

    e. LogMiner

3. Performance
    a. Resource Manager
    b. Wait interface enhancements
    c. End to end application testing
    d. SQL tuning advisor
4. Business Intelligence/Data Warehousing
    a. Materialized view refresh and query rewrite
    b. Asynchronous change data capture
    c. Materialized view refresh and query rewrite
5. Application Development
    a. SQL
    b. PLSQL

Tying everything together is Oracle Enterprise Manager (OEM) which has also been enhanced for Oracle 10g.

OEM in Oracle 10g is installed at the same time as the database and now has a web-based architecture. This eliminates the need to install a separate application to access the OEM console and means the console can also be accessed from a web-enabled PDA via EM2GO. OEM also provides fully functional administration and monitoring after database creation.

The main enhancements to Enterprise Manager for Oracle 10g are in terms of improved ease of use, reduced implementation costs and management of the whole Oracle environment including Oracle Application Server, business applications, storage, network, operating system and hardware.

# 4. FEASIBILITY STUDY

**Feasibility studies** aim to objectively and rationally uncover the strengths and weaknesses of the existing business or proposed venture, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for success. In its simplest term, the two criteria to judge feasibility are cost required and value to be attained. As such, a well-designed feasibility study should provide a historical background of the business or project, description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, feasibility studies precede technical development and project implementation.

*Five common factors (TELOS)*

## 4.1 TECHNOLOGY AND SYSTEM FEASIBILITY

The assessment is based on an outline design of system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. in order to estimate whether the new system will perform adequately or not. Technological feasibility is carried out to determine whether the company has the capability, in terms of software, hardware, personnel and expertise, to handle the completion of the project. When writing a feasibility report the following should be taken to consideration:

- A brief description of the business
- The part of the business being examined
- The human and economic factor
- The possible solutions to the problems

At this level, the concern is whether the proposal is both technically and legally feasible (assuming moderate cost).

## 4.2 ECONOMIC FEASIBILITY

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefits analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action.

Cost-based study: It is important to identify cost and benefit factors, which can be categorized as follows: 1. Development costs; and 2. Operating costs. This is an analysis of the costs to be incurred in the system and the benefits derivable out of the system.Time-based study: This is an analysis of the time required to achieve a return on investments. The future value of a project is also a factor.

## 4.3 LEGAL FEASIBILITY

Determines whether the proposed system conflicts with legal requirements, e.g. a data processing system must comply with the local Data Protection Acts.

## 4.4 OPERATIONAL FEASIBILITY

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development

## 4.5 SCHEDULE FEASIBILITY

A project will fail if it takes too long to be completed before it is useful. Typically this means estimating how long the system will take to develop, and if it can be completed in a given time period using some methods like payback period.

Schedule feasibility is a measure of how reasonable the project timetable is. Given our technical expertise, are the project deadlines reasonable? Some projects are initiated with specific deadlines. You need to determine whether the deadlines are mandatory or desirable.

# 5. SYSTEM ANALYSIS

## 5.1 EXISTING SYSTEM

- With the growth in medical data, collecting electronic health records (EHR) is increasingly convenient. Besides, first presented a bio inspired high-performance heterogeneous vehicular telemetric paradigm.
- The data set consists of sentences from Medline abstracts annotated with disease and treatment entities and with three semantic relations between diseases, symptom and treatments. All the way the information is provided as abstracts in form of the documents. It is possible that the abstracts are made largely available to the viewers ,but it is inconvenient to measure the ratio of facts , our research is focused on different representation techniques, different classification models, and most importantly generates improved results with less annotated data.

### 5.1.1 Drawbacks of Existing System

- It is highly in convenient to read and analyze the medical documents and reviews in the health blogs.
- Large text processing is time consuming.
- Prediction requires weights.

## 5.2 PROPOSED SYSTEM

- Our project is focused on different representation **techniques**, different classification models applied on Big data and most importantly generates improved results with less annotated data.
- The tasks addressed in our research are information extraction and relation extraction. The first task identifies and extracts informative sentences on diseases and treatments topics, while the second one performs a finer grained classification of these sentences according to the semantic relations that exists between diseases and treatments.
- For extracting relations, the rules are used to determine if a textual input contains a relation or not. Taking a statistical approach to solve the relation extraction problem

from abstracts, the most used representation technique is bag-of-words.

- The two tasks that are undertaken in this paper provide the basis for the design of an information technology framework that is capable to identify and disseminate healthcare information. The first task identifies and extracts informative sentences on diseases and treatments topics, while the second one performs a finer grained classification of these sentences according to the semantic relations that exists between diseases and treatments

### 5.2.1 Advantages of Proposed System

1. The final product can be envisioned as a browser plug-in or a desktop application that will automatically find and extract the latest medical discoveries related to disease- treatment relations and present them to the user.

2. The product can be developed and sold by companies that do research in Healthcare Informatics, Natural Language Processing, and Machine Learning, and companies that develop tools like Microsoft Health Vault.

3. The value of the product from an e-commerce point of view stands in the fact that it can be used in marketing strategies to show that the information that is presented is trustful (Medline articles) and that the results are the latest discoveries

# 6. SYSTEM DESIGN

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. If the broader topic of product development "blends the perspective of marketing, design, and manufacturing into a single approach to product development, then design is the act of taking the marketing information and creating the design of the product to be manufactured.

Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user.

## 6.1 UML DIAGRAMS

The Unified Modelling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time-to-market. These techniques include component technology, visual programming, patterns and frameworks. Businesses also seek techniques to manage the complexity of systems as they increase in scope and scale. In particular, they recognize the need to solve recurring architectural problems, such as physical distribution, concurrency, replication, security, load balancing and fault tolerance. Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems. The Unified Modelling Language (UML) was designed to respond to these needs. Simply, Systems design refers to the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements which can be done easily through UML diagrams.

**Contents of UML**

In general, a UML diagram consists of the following features:

- **Entities:** These may be classes, objects, users or systems behaviors.
- Relationship Lines that model the relationships between entities in the system.
- **Generalization --** a solid line with an arrow that points to a higher abstraction of the present item.
- **Association --** a solid line that represents that one entity uses another entity as part of its behaviour.
- **Dependency --** a dotted line with an arrowhead that shows one entity depends on the behaviour of another entity.

In this project four basic UML diagrams have been explained

1) Class Diagram
2) Use Case Diagram
3) Sequence Diagram

### 6.1.1 Class Diagram

UML class diagrams model static class relationships that represent the fundamental architecture of the system. Note that these diagrams describe the relationships between classes, not those between specific objects instantiated from those classes. Thus the diagram applies to all the objects in the system.

A class diagram consists of the following features:

- **Classes:** These titled boxes represent the classes in the system and contain information about the name of the class, fields, methods and access specifies. Abstract roles of the Class in the system can also be indicated.

- **Interfaces:** These titled boxes represent interfaces in the system and contain information about the name of the interface and its methods. Relationship Lines that model the relationships between classes and interfaces in the system.
- **Dependency:** A dotted line with an open arrowhead that shows one entity depends on the behaviour of another entity. Typical usages are to represent that one class instantiates another or that it uses the other as an input parameter
- **Aggregation:** Represented by an association line with a hollow diamond at the tail end. An aggregation models the notion that one object uses another object without "owning" it and thus is not responsible for its creation or destruction.
- **Inheritance:** A solid line with a solid arrowhead that points from a sub-class to a super class or from a sub-interface to its super-interface.
- **Implementation:** A dotted line with a solid arrowhead that points from a class to the interface that it implement
- **Composition:** Represented by an association line with a solid diamond at the tail end. A composition models the notion of one object "owning" another and thus being responsible for the creation and destruction of another

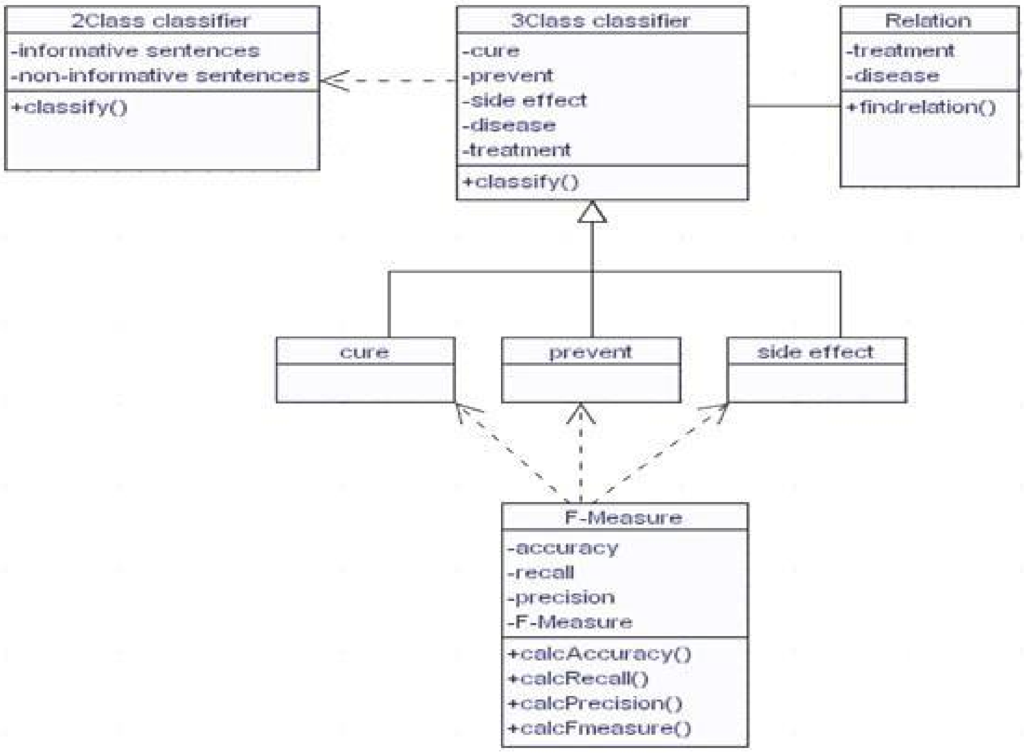


**Figure 1: Class diagram for information extraction**

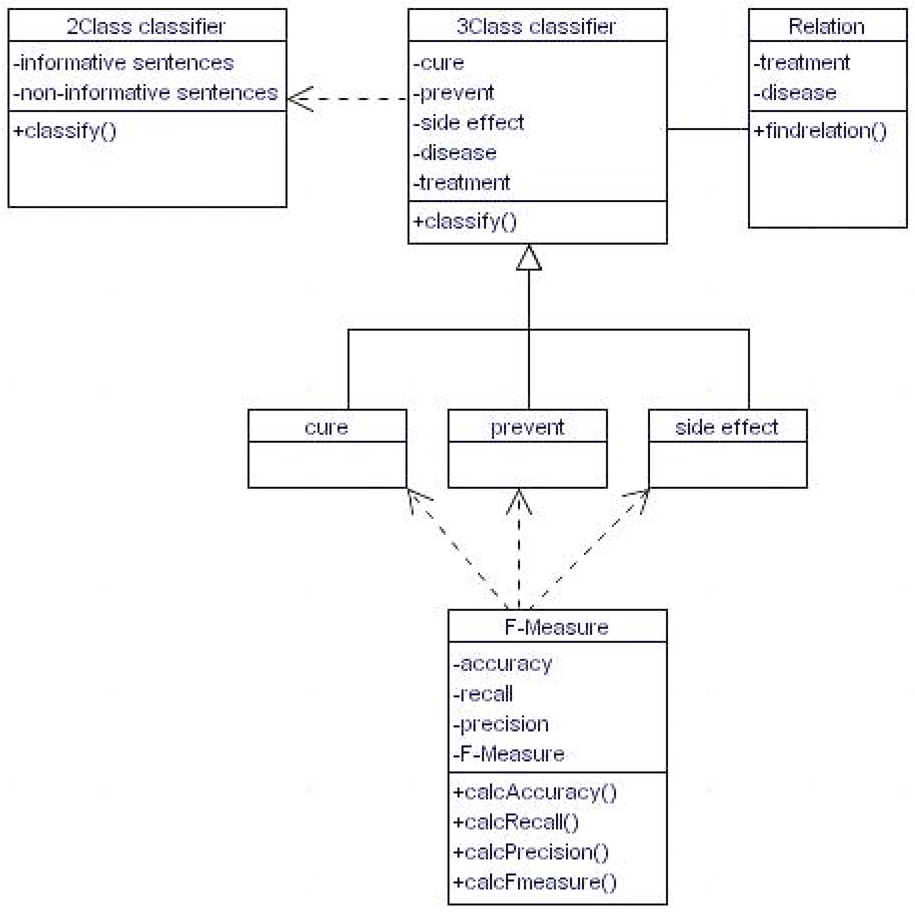


**Figure 2: Class Diagram For Relation Identification**

### 6.1.2 Use Case Diagram:

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms.

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. It consists of a group of elements (for example, classes and interfaces) that can be used together in a way that will have an effect larger than the sum of the separate elements combined. The use case should contain all system activities that have significance to the users. A use case can be thought of as a collection of possible scenarios related to a particular goal, indeed, the use case and goal are sometimes considered to be synonymous.

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

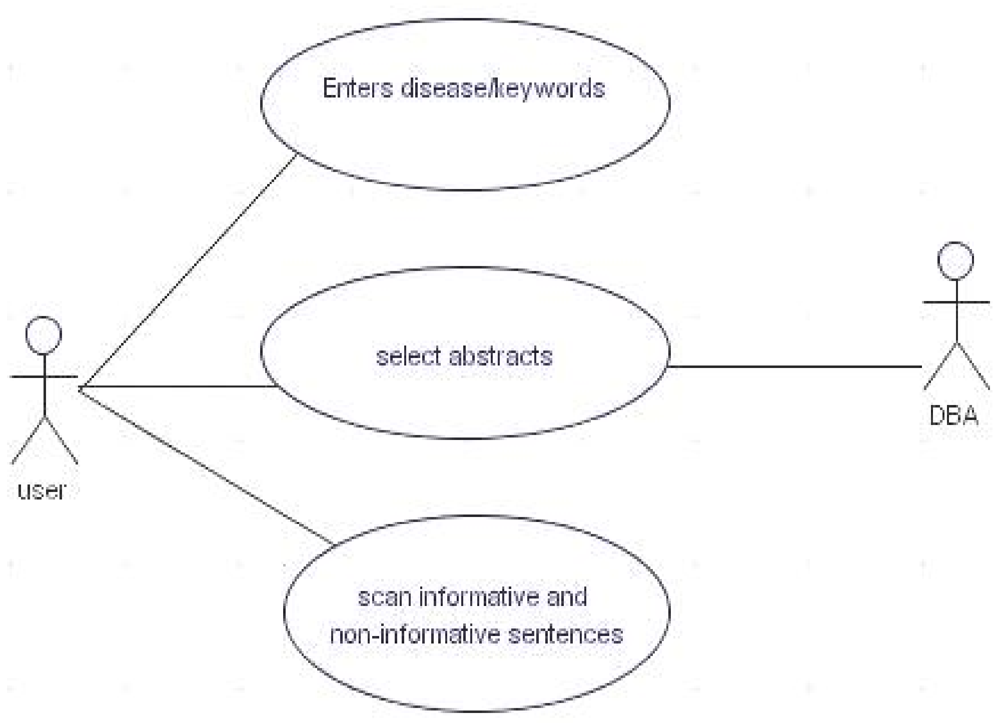


**Figure 3: Use case Diagram for Information Extraction**

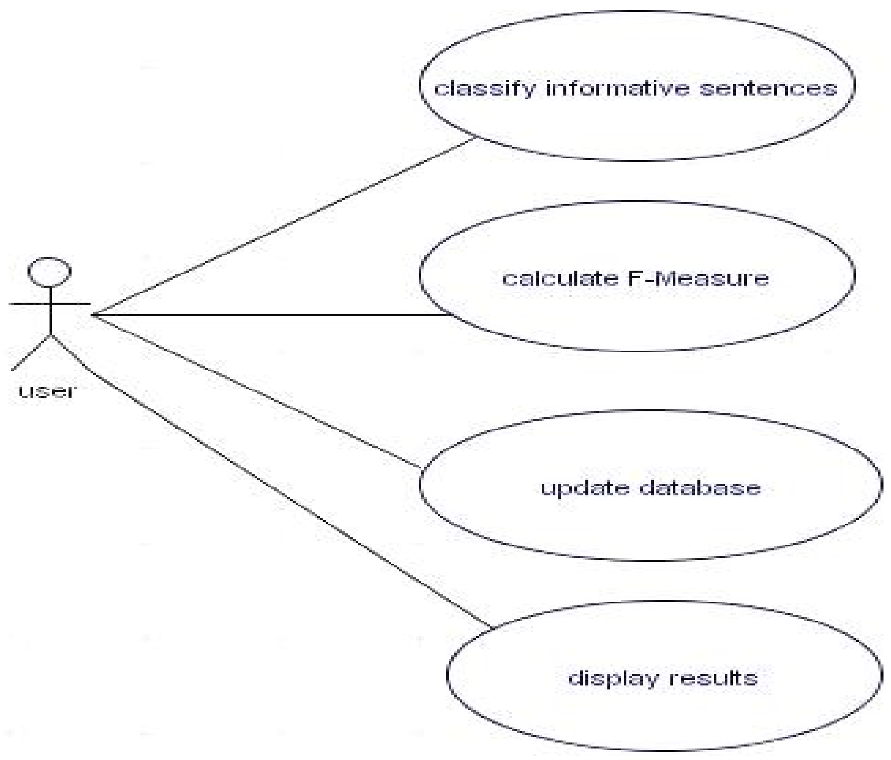


**Figure 4: Use Case Diagram for Relation Identification**

- **Parts of Use cases**

  A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.

- **Actors**

  An actor is a person, organization, or external system that plays a role in one or more interactions with the system.

- **System boundary boxes (optional)**

  A rectangle is drawn around the use cases, called the system boundary box, to indicate the scope of system. Anything within the box represents functionality that is in scope and anything outside the box is not **Relationships.**

- **Include**

In one form of interaction, a given use case may include another. "Include is a Directed Relationship between two use cases, implying that the behaviour of the included use case is inserted into the behaviour of the including use case".

The first use case often depends on the outcome of the included use case. This is useful for extracting truly common behaviours from multiple use cases into a single description. The notation is a dashed arrow from the including to the included use case, with the label "«include»". This usage resembles a macro expansion where the included use case behaviour is placed inline in the base use case behaviour. There are no parameters or return values. To specify the location in a flow of events in which the base use case includes the behaviour of another, you simply write include followed by the name of use case you want to include, as in the following flow for track order.

- **Extend**

In another form of interaction, a given use case (the extension) may extend another. This relationship indicates that the behaviour of the extension use case may be inserted in the extended use case under some conditions. The notation is a dashed arrow from the extension to the extended use case, with the label "«extend»". The notes or constraints may be associated with this relationship to illustrate the conditions under which this behaviour will be executed. Modellers use the «extend» relationship to indicate use cases that are "optional" to the base use case. Depending on the modeller's approach "optional" may mean "potentially not executed with the base use case" or it may mean "not required to achieve the base use case goal".

- **Generalization**

In the third form of relationship among use cases, a generalization/ specialization relationship exists. A given use case may have common behaviours, requirements, constraints, and assumptions with a more general use case. In this case, describe them once, and deal with it in the same way, describing any differences in the specialized cases. The notation is a solid line ending in a hollow triangle drawn from the specialized to the more general use case (following the standard generalization notation).

➢ **Associations**

Associations between actors and use cases are indicated in use case diagrams by solid lines. An association exists whenever an actor is involved with an interaction described by a use case. Associations are modelled as lines connecting use cases and actors to one another, with an optional arrowhead on one end of the line. The arrowhead is often used to indicate the direction of the initial invocation of the relationship or to indicate the primary actor within the use case. The arrowheads imply control flow and should not be confused with data flow.

➢ **Steps To Draw Use Cases**

- Identifying Actor
- Identifying Use cases
- Review your use case for completeness

### 6.1.2 Sequence Diagram

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct  of a Message Sequence Chart. A Sequence diagram depicts the sequence of actions that occur in a system. The invocation of methods in each object, and the order in which the invocation occurs is captured in a Sequence diagram. This makes the Sequence diagram a very useful tool to easily represent the dynamic behaviour of a system.

➢ **Elements of sequence diagram**

The sequence diagram is an element that is used primarily to showcase  the interaction that occurs between multiple objects. This interaction will be shown over certain period of time. Because of this, the first symbol that is used is one that symbolizes the object.

➢ **Lifeline**

A lifeline will generally be generated, and it is a dashed line that sits vertically, and the top will be in the form of a rectangle. This rectangle is used to indicate both the instance and the class. If the lifeline must be used to denote an object, it will be underlined.

➢ **Messages**

To showcase an interaction, messages will be used. These messages will come in the form of horizontal arrows, and the messages should be written on top of the arrows. If the arrow has a full head, and it's solid, it will be called a synchronous call. If the solid arrow has a stick head, it will be an asynchronous call. Stick heads with dash arrows are used to represent return messages.

➢ **Objects**

Objects will also be given the ability to call methods upon themselves, and they can add net activation boxes. Because of this, they can communicate with others to show multiple levels of processing. Whenever an object is eradicated or erased from memory, the "X" will be drawn at the lifeline's top, and the dash line will not be drawn beneath it. This will often occur as a result of a message. If a message is sent from the outside of the diagram, it can be used to define a message that comes from a circle that is filled in. Within a UML based model, a Super step is a collection of steps which result from outside stimuli.

**Steps to Create a Sequence Diagram**

- Set the context for the interaction, whether it is a system, subsystem, operation or class.
- Set the stage for the interaction by identifying which objects play a role in interaction.
- Set the lifetime for each object.
- Start with the message that initiates the interaction.
- Visualize the nesting of messages or the points in time during actual computation.

- Specify time and space constraints, adorn each message with timing mark and attach suitable time or space constraints.
- Specify the flow of control more formally, attach pre and post conditions to each message.

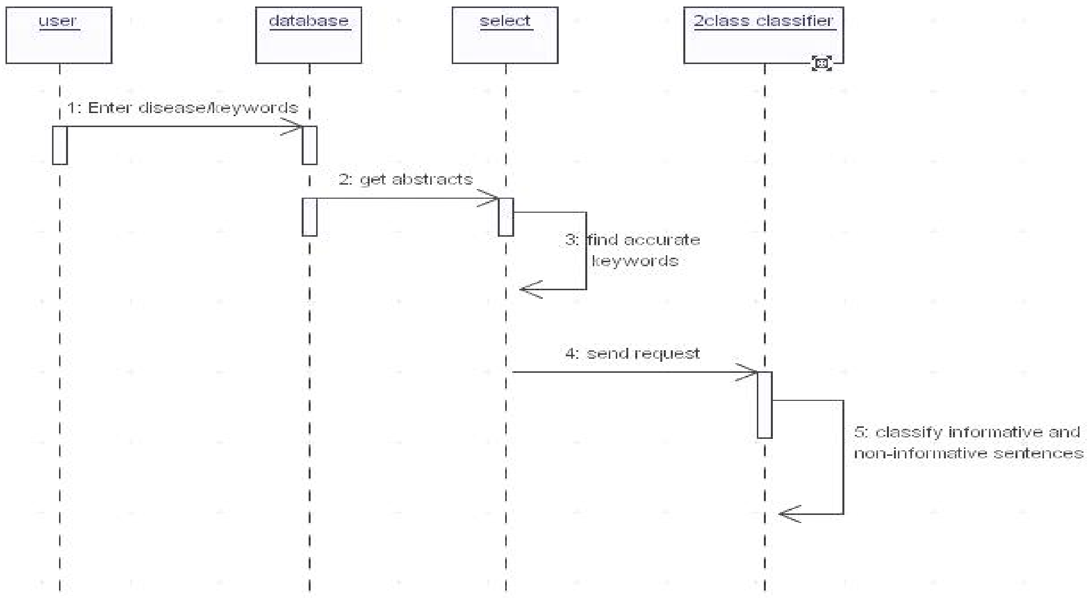

user
database
select
2class classifier
1: Enter disease/keywords
2: get abstracts
3: find accurate keywords
4: send request
5: classify informative and non-informative sentences

**Figure 5: Sequence Diagram For Information Extraction**

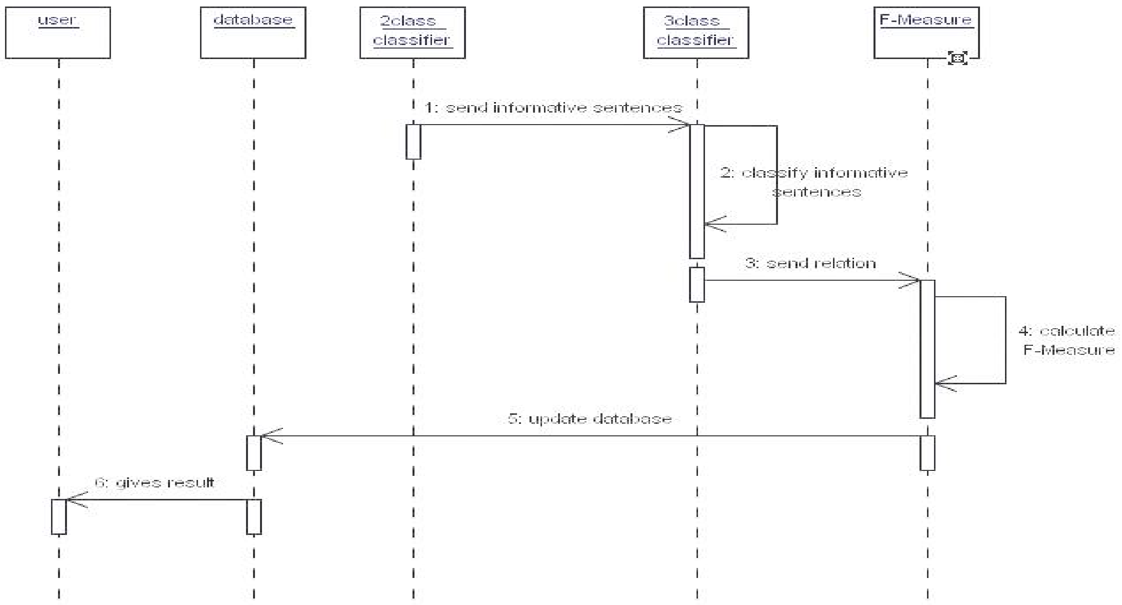
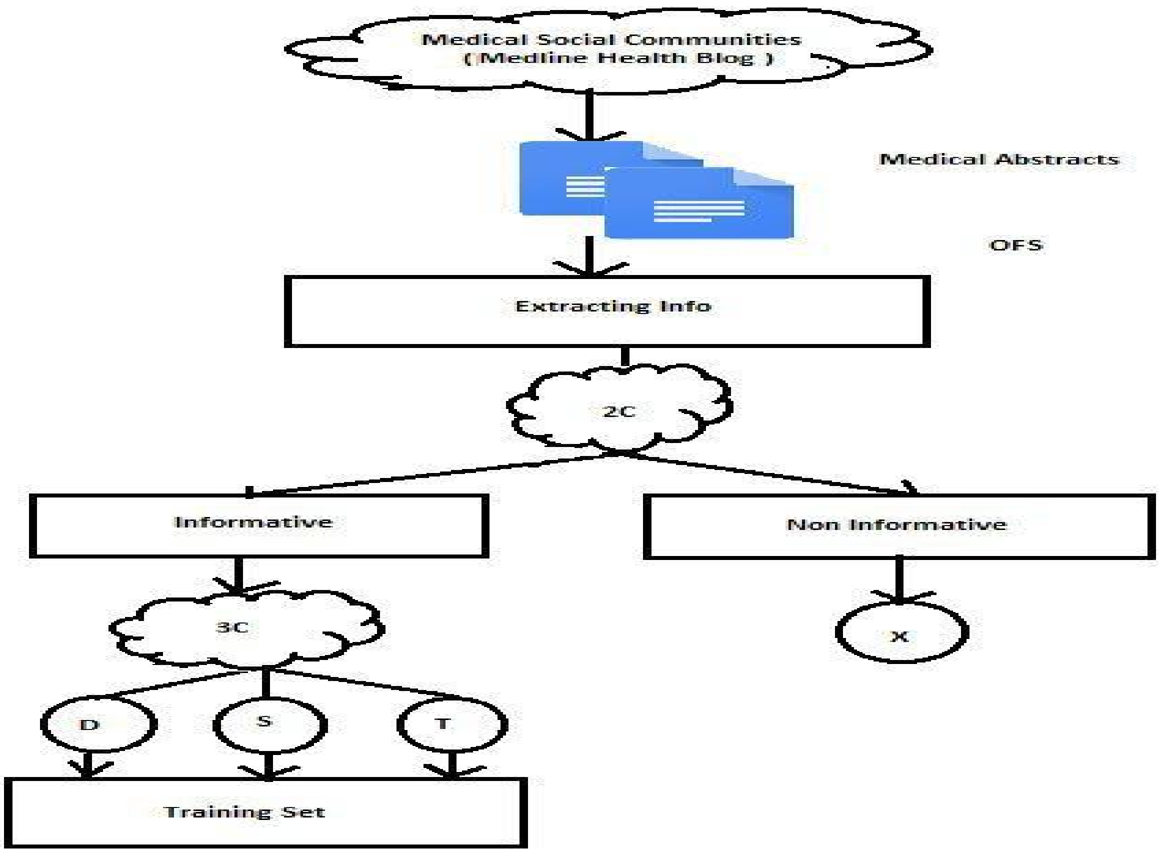


**Figure 6: Sequence Diagram For Relation Identification**

## 6.2 ARCHITECTURAL DESIGN



# 7. SYSTEM IMPLEMENTATION

## 7.1 IMPLEMENTATION PROCESS

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

## 7.2 MODULES DESCRIPTION

### Module 1: Information Extraction

The first task (task 1 or sentence selection) identifies sentences from Medline published abstracts that talk about diseases and treatments. The task is similar to a scan of sentences contained in the abstract of an article in order to present to the user-only sentences that are identified as containing relevant information (disease treatment information).

### Module 2: Relation Identification

The second task (task 2 or relation identification) has a deeper semantic dimension and it is focused on identifying disease-treatment relations in the sentences already selected as being informative (e.g., task 1 is applied first). We focus on three relations: Cure, Prevent, and Side Effect, a subset of the eight relations that the corpus is annotated with.

## Classification Algorithms and Data Representations

As classification algorithms, we use a set of six representative models: decision-based models (Decision trees), probabilistic models (Naive Bayes (NB) and Complement Naive Bayes (CNB), which is adapted for text with imbalanced class distribution), adaptive learning (Ada- Boost), a linear classifier (support vector machine (SVM) with polynomial kernel), and a classifier that always predicts the majority class in the training data (used as a baseline).

## Feature Selection For Chronic Disease Prediction

Feature selection, also known as Variable Selection is an extensively used data preprocessing technique in data mining which is basically used for reduction of data by eliminating insignificant and superfluous attributes from any dataset. Moreover, this technique enhances the comprehensibility of data, facilitates better visualization of data, reduces training time of learning algorithm and improves the performance of prediction.

There exist numerous applications of relevant feature identification techniques in healthcare sector. Filter methods, wrapper methods, ensemble methods and embedded methods are some of the popularly used techniques used for variable selection. In recent years, most of the authors are focusing on hybrid approaches used for feature selection.

Before any model is applied to the data, it is always better to remove noisy and inconsistent data to get more accurate results in less time. Reducing the dimensionality of a dataset is of paramount importance in real-world applications. Moreover, if most important features are selected, the complexity decreases exponentially.

In recent years, various feature identification approaches have been applied on healthcare datasets to get more valuable information. The utilization of feature selection methods is done on clinical databases for the prediction of numerous chronic diseases like diabetes, heart disease, strokes, hypertension, thalassemia etc. Various learning algorithms work efficiently and give more accurate results if the data contains more significant and non-redundant attributes. As the medical datasets contains large number of redundant & irrelevant features, an efficient feature selection technique is needed to extract interesting features relevant to the disease.

# 8. TESTING

A "program unit" stands for a routine or a collection of routines implemented by an individual programmer. It might even be a stand-alone program or a functional unit a larger program.

## 8.1 UNIT TESTING

Unit testing is performed prior to integration of the unit into a larger system. It is like coding and debugging ->unit testing ->integration testing. A program unit must be tested for functional tests, performance tests, stress tests and structure tests.

Functional tests refer to executing the code with standard inputs, for which the results will occur within the expected boundaries. Performance test determines the execution time spent in various parts of the unit, response time, device utilization and throughput. Performance testing will help the tuning of the system.

Stress tests drive the system to its limits. They are designed to internationally break the unit. Structure tests verify logical execution along different execution paths. Functional, performance and stress tests are collectively known as "black box testing". Structure testing is referred to as "white box" or "glass box" testing. Program errors can be classified as missing path errors, computational errors and domain errors.

Even if it looks like all possible execution paths have been tested, there might still exist some more paths. A missing path error occurs, when a branching statement and the associated computations are accidentally omitted. Missing paths can be detected only by functional specifications. A domain error occurs when a program traverses the wrong path because of an incorrect predicate in a branching statement. When a test case fails to detect a computational error there is said to be a coincidental error.

**Debugging**

Debugging is eliminating the cause of known errors. Commonly used debugging techniques are induction, deduction and backtracking. Debugging by induction involves the following steps:

- Collect all the information about test details and test results
- Look for patterns
- Form one or more hypotheses and rank/classify them.
- Prove/disprove hypotheses. Re examine
- Implement appropriate corrections
- Verify the corrections. Re run the system and test again until satisfactory
- Debugging by deduction involves the following steps:
- List possible causes for observed failure.
- Use the available information to eliminate various hypotheses.
- Prove/disprove the remaining hypotheses.
- Determine the appropriate correction.
- Carry out the corrections and verify.

Debugging by backtracking involves working backward in the source code from point where the error was observed. Run additional test cause and collect more information.

## 8.2 INTEGRATION TESTING

Integration testing strategies include bottom-up (traditional), top-down and sandwich strategies. Bottom-up integration consists of unit testing, followed by testing entire system. Unit testing tries to discover errors in modules. Modules are tested independently in an artificial environment known as “test harness”. Test harnesses provide data environments and calling sequences for the routines and subsystem that are being tested in isolation.

Disadvantages of bottom-up testing include that harness preparation, which can sometimes take about 50% or more of the coding and debugging effort for a smaller product. After testing all the modules independently and in isolation, they are linked and executed in one single integration run. This is known as "Big bang" approach to integration testing. Isolating sources of errors is difficult in "big bang" approach.

Top-down integration starts with main routine and one or two immediately next lower level routines. After a through checking the top level becomes a test harness to its immediate subordinate routines. Top-down integration offers the following advantages. System integration is distributed throughout the implementation phase. Modules are integrated as they are developed.

- Top-level interfaces are first test.
- Top-level routines provide a natural test harness for lower-level routines.
- Errors are localized to the new modules and interfaces that are being added.

Though top-down integrations seem to offer better advantages, it may not be applicable in certain situations. Sometimes it may be necessary to test certain low-level modules first. In such situations, a sandwich strategy is preferable. Sandwich integration is mostly top-down, but bottom-up techniques are used on some modules and sub systems. This mixed approach retains the advantages of both strategies.

## 8.3 SYSTEM TESTING

System testing involves two activities: Integration testing and Acceptance testing. Integration strategy stresses on the order in which modules are written, debugged and unit tested. Acceptance test involves functional tests, performance tests and stress tests to verify requirements fulfilment. System checking checks the interface, decision logic, control flow, recovery procedures and throughput, capacity and timing characteristics of the entire system.

## 8.4 ACCEPTANCE TESTING

Acceptance testing involves planning and execution of functional tests, performance tests and stress tests in order to check whether the system implemented satisfies the requirements specifications. Quality assurance people as well as customers may simultaneously develop acceptance tests and run them. In additional to functional and performance tests, stress test are performed to determine the limits/limitations of the system developed. For example, a compiler may be tested for its symbol table overflows or a real-time system may be tested for multiple interrupts of different/same priorities.

Acceptance test tools include a test coverage analyzer, and a coding standards checker. Test coverage analyzer records the control paths followed for each test case. A timing analyzer reports the time spent in various regions of the source code under different test cases. Coding standard are stated in the product requirements. Manual inspection is usually not an adequate mechanism from detecting violations of coding standards.

## Testing Objectives

Testing is a process of execution a program with the intent of finding on errors. A good test is on that has a high probability of finding an undiscovered errors. Testing is vital to the success of the system. System testing is the state of implementation, which ensures that the system works accurately before live operations commence. System testing makes a logical assumption that the system is correct and that the system is correct and that the goals are successfully achieved.

## Effective Testing Prerequisites

### Integration testing

An overall test plan for the project is prepared before the start of coding.

### Validation testing

This project will be tested under this testing sample data and produce the correct sample
output.

### Recovery testing

This project will be tested under this testing using correct data input and its product and the correct valid output without any errors.

### Security testing

This project contains password to secure the data.

### Test Data and Input

Taking various types of data we do the above testing. Preparation of test data plays a vital role in system testing. After preparing the test data the system under study is treated using the test data.While testing the system by using the above testing and correction methods. The system has been verified and validated by running with both.

- Run with live data
- Run with test data

### Run With Test Data

In the case the system was run with some sample data. Specification testing was also done for each conditions or combinations for conditions.

### Run With Live Data

The system was tested with the data of the old system for a particular period. Then the new reports were verified with the old one.

# 9. SAMPLE SOURCE CODE

**Abstract.java**

```
 package desktopapplication1;
/**

*

  * @author Ranganath

  */

 public class Abstract {

    public String toString()
    {
    return id+"";
    }
 private int id;
private      String
url; private int rank;
           private String
           xmlText; private
           String disease;
        private String Text
        public Abstract() {
                    id = 0; url = null;

                    rank = 0; xmlText = null; disease = null;

                    // TODO Auto-generated constructor stub
           }

           public Abstract(int id, String url, int rank, String xmlText, String disease) {

                    super(); this.id = id; this.url = url;

                    this.rank = rank; this.xmlText = xmlText; this.disease = disease;

           }

           public int getId() {

                    return id;

           }

           public void setId(int id) {
                    this.id = id;

           }

           public String getUrl() { return url;

           }

           public void setUrl(String url) { this.url = url;

           }

           public int getRank() {

                    return rank;

           }

           public void setRank(int rank) { this.rank = rank;

           }

           public String getXmlText() { return xmlText;

           }
```

```
            public void setXmlText(String xmlText) { this.xmlText = xmlText;

            }

            public String getDisease() { return disease;
            }

            public void setDisease(String disease) { this.disease = disease;

            }
        /**

         * @return the Text

         */

        public String getText() { return Text;

        }
      /**

         * @param Text the Text to set

         */

        public void setText(String Text) { this.Text = Text;

        }

}
```

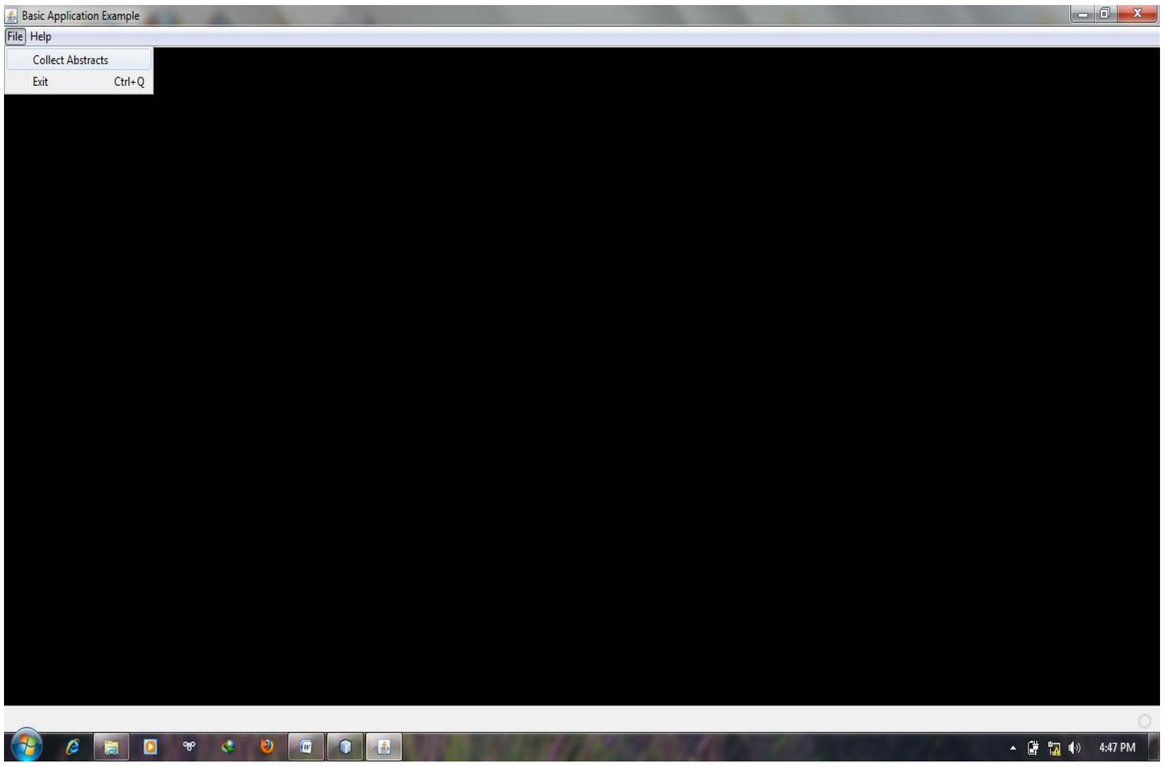
**AbstractFactory.java**

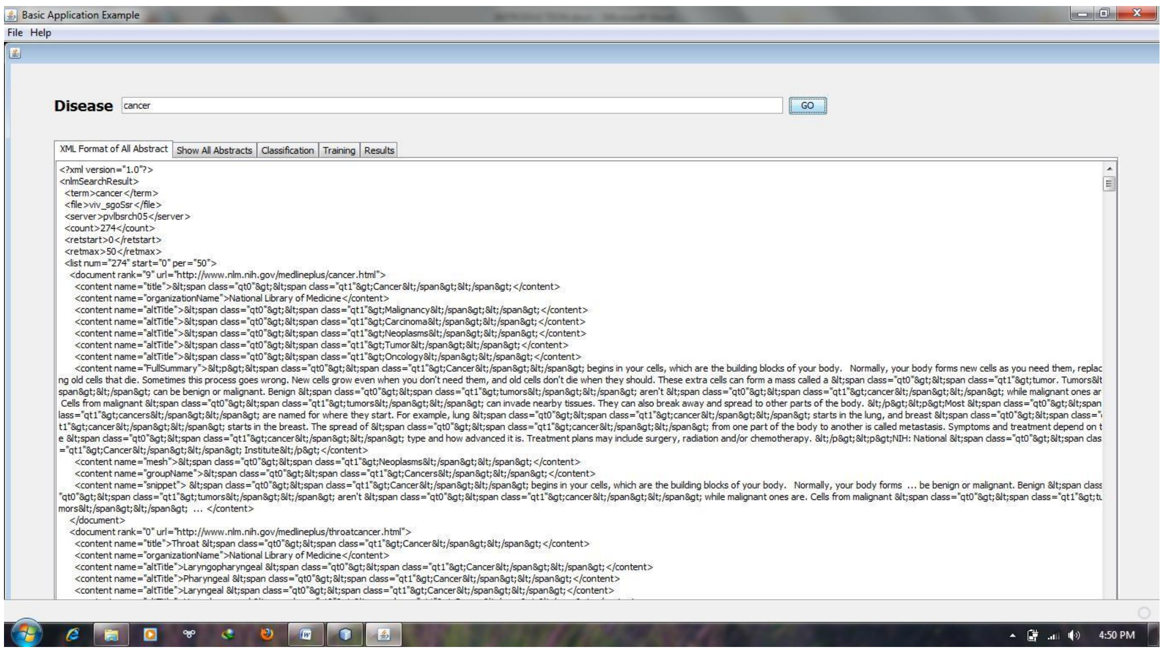
```
o change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package desktopapplication1;
 import java.sql.Connection;    import java.sql.DriverManager;
import java.sql.PreparedStatement; import java.sql.ResultSet;
import java.util.TreeMap;
/**
 *
 * @author Ranganath
 */
public class AbstractFactory
{
PreparedStatement ps_I,ps_U,ps_D,ps_F1,ps_F2,ps_F3; public AbstractFactory()throws
Exception
{
Class.forName(DataBase.driver_class);
Connection
con=DriverManager.getConnection(DataBase.connectionURL,DataBase.user,DataBase.
psw);
ps_I=con.prepareStatement("insert into
abstracts(ABSTRACT_URL,ABSTRACT_RANK,ABSTRACT_XMLTEXT,DIS
EASE)
values(?,?,?,?)");

  ps_U=con.prepareStatement("update Abstracts set
abstract_url=?,abstract_rank=?,abstract_xmlText=?,disease=?,abstract_text=?
where abstract_id=?");

       ps_D=con.prepareStatement("delete from Abstracts where id=?");

        ps_F1=con.prepareStatement("select
id,url,rank,xmlText,disease,abstract_text from Abstracts");

         ps_F2=con.prepareStatement("select
id,url,rank,xmlText,disease,abstract_text from Abstracts where id=?");

          ps_F3=con.prepareStatement("select
ABSTRACT_id,ABSTRACT_URL,ABSTRACT_rank,ABSTRACT_xmlText,disease,abs
tract_t ext from Abstracts where disease=?");

}
public boolean create(Abstract a)throws Exception

{

boolean done=false;

ps_I.setString(1,a.get

Url());

ps_I.setInt(2,a.getRa

nk());
if(a.getXmlText().length()>=4000)

   ps_I.setString(3,

a.getXmlText().substring(0,4000)); else

ps_I.setString(3,a.getXmlT
  ext());

  ps_I.setString(4,a.getDisea

  se());

  if(ps_I.executeUpdate()>0)

     don

  e=true;

  return

  done;

  }

   public boolean store(Abstract e )throws Exception

   {

   boolean done=false;


 ps_U.setString(1, e.getUrl());

ps_U.setInt(2, e.getRank());

ps_U.setString(3,

e.getXmlText());

ps_U.setString(4,

e.getDisease());

ps_U.setString(5, e.getText());

ps_U.setInt(6,

e.getId());

if(ps_U.executeUpdate()
>0)

            done=true;
  return done;

}
   public boolean remove(int id )throws Exception

   {

   boolean done=false;

  ps_D.setInt(1,id);

  if(ps_D.executeUpdate()

  >0)

            done=true;

  return done;
 }

 public boolean remove(Abstract e)throws Exception

   {

return remove(e.getId());

   }

   public TreeMap<Integer,Abstract> findAll()throws Exception

  {


 TreeMap<Integer,Abstract> all=new

TreeMap<Integer,Abstract>(); ResultSet

rs=ps_F1.executeQuery();

   while(rs.next())

   {
```

```
   Abstract e=getAbstracts(rs); all.put(e.getId(), e);

  }

  rs.close();

  return all;
 }

public Abstract findById(int id)throws Exception

{

 Abstract e=null; ps_F2.setInt(1, id);

 ResultSet rs=ps_F2.executeQuery(); rs.next();

 e=getAbstracts(rs);

 rs.close();

  return e;

 }

public TreeMap<Integer,Abstract> findByDesease(String desease)throws Exception

{

 TreeMap<Integer,Abstract> all=new TreeMap<Integer,Abstract>(); ps_F3.setString(1,

 desease);

 ResultSet rs=ps_F3.executeQuery(); while(rs.next())

 {

   Abstract e=getAbstracts(rs); all.put(e.getId(), e);

  }

 rs.close(); return all;
```

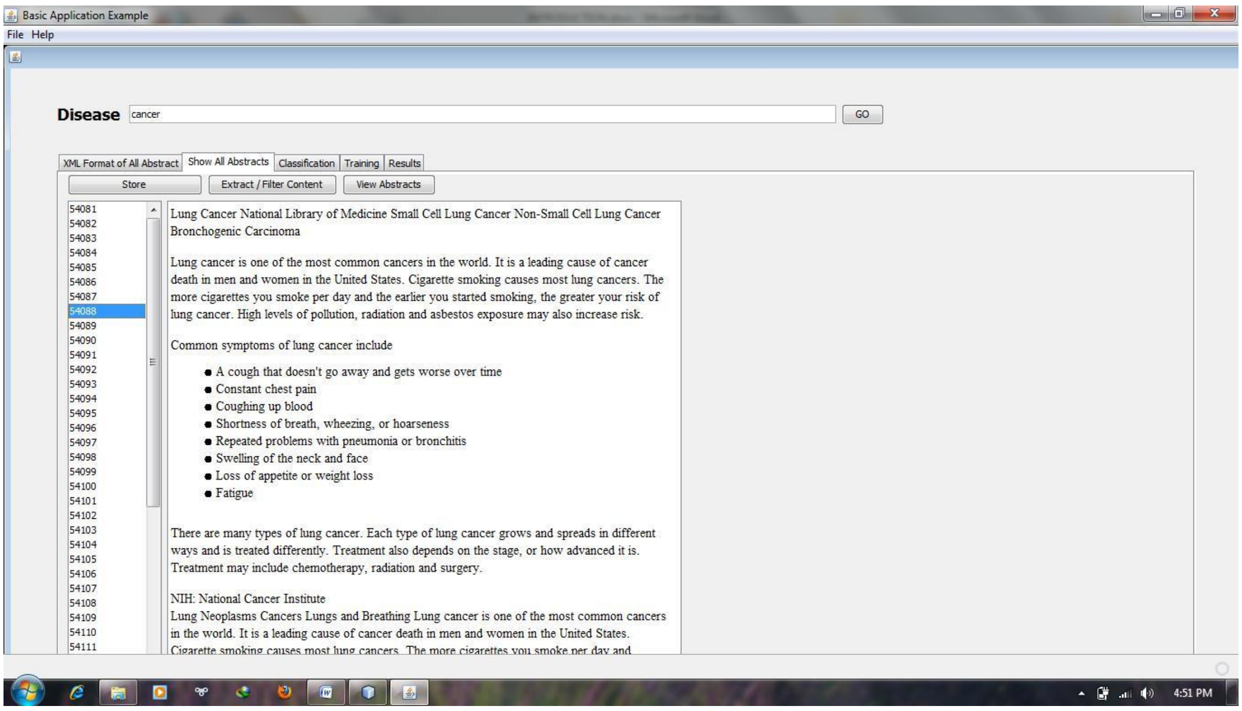
```
  }

public Abstract getAbstracts(ResultSet rs)throws Exception

   {

      Abstract e=new Abstract(); e.setId(rs.getInt(1)); e.setUrl(rs.getString(2));

      e.setRank(rs.getInt(3));
      e.setXmlText(rs.getString(4)); e.setDisease(rs.getString(5)); e.setText(rs.getString(6));

   // e.setDname(new DeptManager().findBydisease(e.getdisease()).getDname()); return e;

   }

}
```

# 10. SCREEN LAYOUTS





```
<?xml version="1.0"?>
<nlmSearchResult>
  <term>cancer</term>
  <file>viv_sgoSsr</file>
  <server>pvlbsrch05</server>
  <count>274</count>
  <retstart>0</retstart>
  <retmax>50</retmax>
  <list num="274" start="0" per="50">
    <document rank="9" url="http://www.nlm.nih.gov/medlineplus/cancer.html">
      <content name="title">&lt;span class="qt0"&gt;&lt;span class="qt1"&gt;Cancer&lt;/span&gt;&lt;/span&gt;</content>
      <content name="organizationName">National Library of Medicine</content>
      <content name="altTitle">&lt;span class="qt0"&gt;&lt;span class="qt1"&gt;Malignancy&lt;/span&gt;&lt;/span&gt;</content>
      <content name="altTitle">&lt;span class="qt0"&gt;&lt;span class="qt1"&gt;Carcinoma&lt;/span&gt;&lt;/span&gt;</content>
      <content name="altTitle">&lt;span class="qt0"&gt;&lt;span class="qt1"&gt;Neoplasms&lt;/span&gt;&lt;/span&gt;</content>
      <content name="altTitle">&lt;span class="qt0"&gt;&lt;span class="qt1"&gt;Tumor&lt;/span&gt;&lt;/span&gt;</content>
      <content name="altTitle">&lt;span class="qt0"&gt;&lt;span class="qt1"&gt;Oncology&lt;/span&gt;&lt;/span&gt;</content>
      <content name="FullSummary">&lt;p&gt;&lt;span class="qt0"&gt;&lt;span class="qt1"&gt;Cancer&lt;/span&gt;&lt;/span&gt; begins in your cells, which are the building blocks of your body.   Normally, your body forms new cells as you need them, replac
ng old cells that die. Sometimes this process goes wrong. New cells grow even when you don't need them, and old cells don't die when they should. These extra cells can form a mass called a &lt;span class="qt0"&gt;&lt;span class="qt1"&gt;tumor. Tumors&lt
span&gt;&lt;/span&gt; can be benign or malignant. Benign &lt;span class="qt0"&gt;&lt;span class="qt1"&gt;tumors&lt;/span&gt;&lt;/span&gt; aren't &lt;span class="qt0"&gt;&lt;span class="qt1"&gt;cancer&lt;/span&gt;&lt;/span&gt; while malignant ones ar
 Cells from malignant &lt;span class="qt0"&gt;&lt;span class="qt1"&gt;tumors&lt;/span&gt;&lt;/span&gt; can invade nearby tissues. They can also break away and spread to other parts of the body. &lt;/p&gt;&lt;p&gt;Most &lt;span class="qt0"&gt;&lt;span
lass="qt1"&gt;cancers&lt;/span&gt;&lt;/span&gt; are named for where they start. For example, lung &lt;span class="qt0"&gt;&lt;span class="qt1"&gt;cancer&lt;/span&gt;&lt;/span&gt; starts in the lung, and breast &lt;span class="qt0"&gt;&lt;span class="
t1"&gt;cancer&lt;/span&gt;&lt;/span&gt; starts in the breast. The spread of &lt;span class="qt0"&gt;&lt;span class="qt1"&gt;cancer&lt;/span&gt;&lt;/span&gt; from one part of the body to another is called metastasis. Symptoms and treatment depend on t
e &lt;span class="qt0"&gt;&lt;span class="qt1"&gt;cancer&lt;/span&gt;&lt;/span&gt; type and how advanced it is. Treatment plans may include surgery, radiation and/or chemotherapy. &lt;/p&gt;&lt;p&gt;NIH: National &lt;span class="qt0"&gt;&lt;span clas
="qt1"&gt;Cancer&lt;/span&gt;&lt;/span&gt; Institute&lt;/p&gt;</content>
      <content name="mesh">&lt;span class="qt0"&gt;&lt;span class="qt1"&gt;Neoplasms&lt;/span&gt;&lt;/span&gt;</content>
      <content name="groupName">&lt;span class="qt0"&gt;&lt;span class="qt1"&gt;Cancers&lt;/span&gt;&lt;/span&gt;</content>
      <content name="snippet"> &lt;span class="qt0"&gt;&lt;span class="qt1"&gt;Cancer&lt;/span&gt;&lt;/span&gt; begins in your cells, which are the building blocks of your body.   Normally, your body forms  ... be benign or malignant. Benign &lt;span class
"qt0"&gt;&lt;span class="qt1"&gt;tumors&lt;/span&gt;&lt;/span&gt; aren't &lt;span class="qt0"&gt;&lt;span class="qt1"&gt;cancer&lt;/span&gt;&lt;/span&gt; while malignant ones are. Cells from malignant &lt;span class="qt0"&gt;&lt;span class="qt1"&gt;tu
mors&lt;/span&gt;&lt;/span&gt;  ... </content>
    </document>
    <document rank="0" url="http://www.nlm.nih.gov/medlineplus/throatcancer.html">
      <content name="title">Throat &lt;span class="qt0"&gt;&lt;span class="qt1"&gt;Cancer&lt;/span&gt;&lt;/span&gt;</content>
      <content name="organizationName">National Library of Medicine</content>
      <content name="altTitle">Laryngopharyngeal &lt;span class="qt0"&gt;&lt;span class="qt1"&gt;Cancer&lt;/span&gt;&lt;/span&gt;</content>
      <content name="altTitle">Pharyngeal &lt;span class="qt0"&gt;&lt;span class="qt1"&gt;Cancer&lt;/span&gt;&lt;/span&gt;</content>
      <content name="altTitle">Laryngeal &lt;span class="qt0"&gt;&lt;span class="qt1"&gt;Cancer&lt;/span&gt;&lt;/span&gt;</content>
```



Lung Cancer National Library of Medicine Small Cell Lung Cancer Non-Small Cell Lung Cancer Bronchogenic Carcinoma

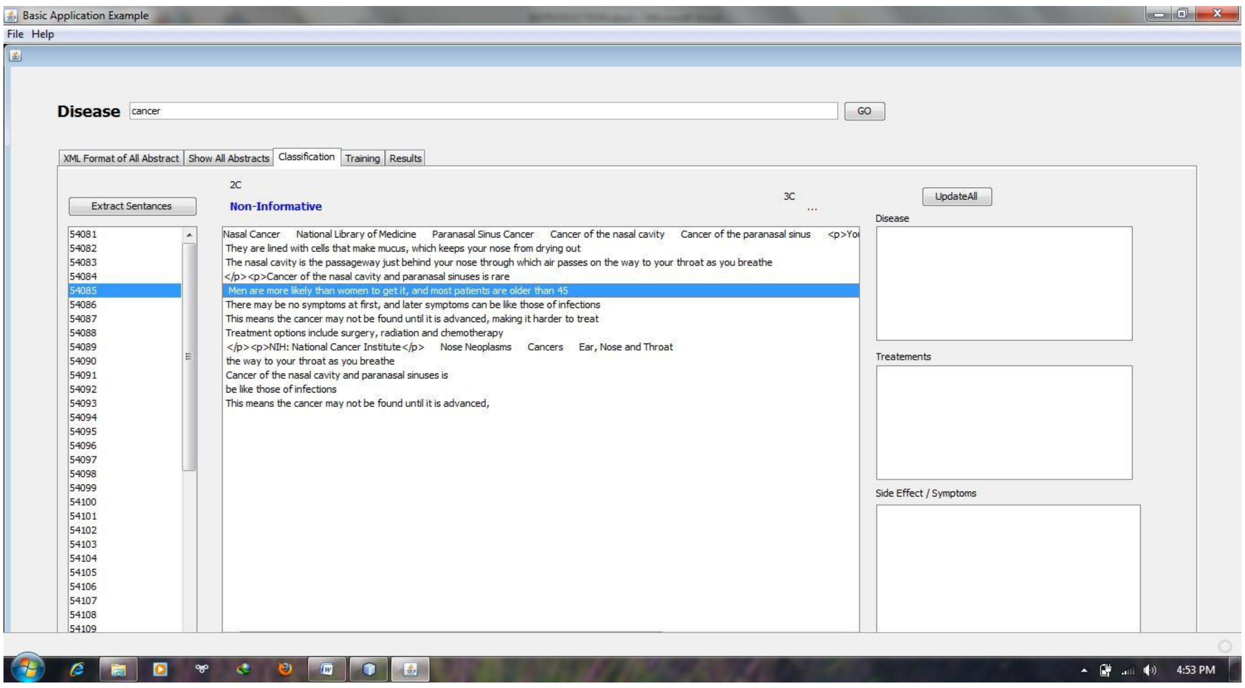
Lung cancer is one of the most common cancers in the world. It is a leading cause of cancer death in men and women in the United States. Cigarette smoking causes most lung cancers. The more cigarettes you smoke per day and the earlier you started smoking, the greater your risk of lung cancer. High levels of pollution, radiation and asbestos exposure may also increase risk.

Common symptoms of lung cancer include

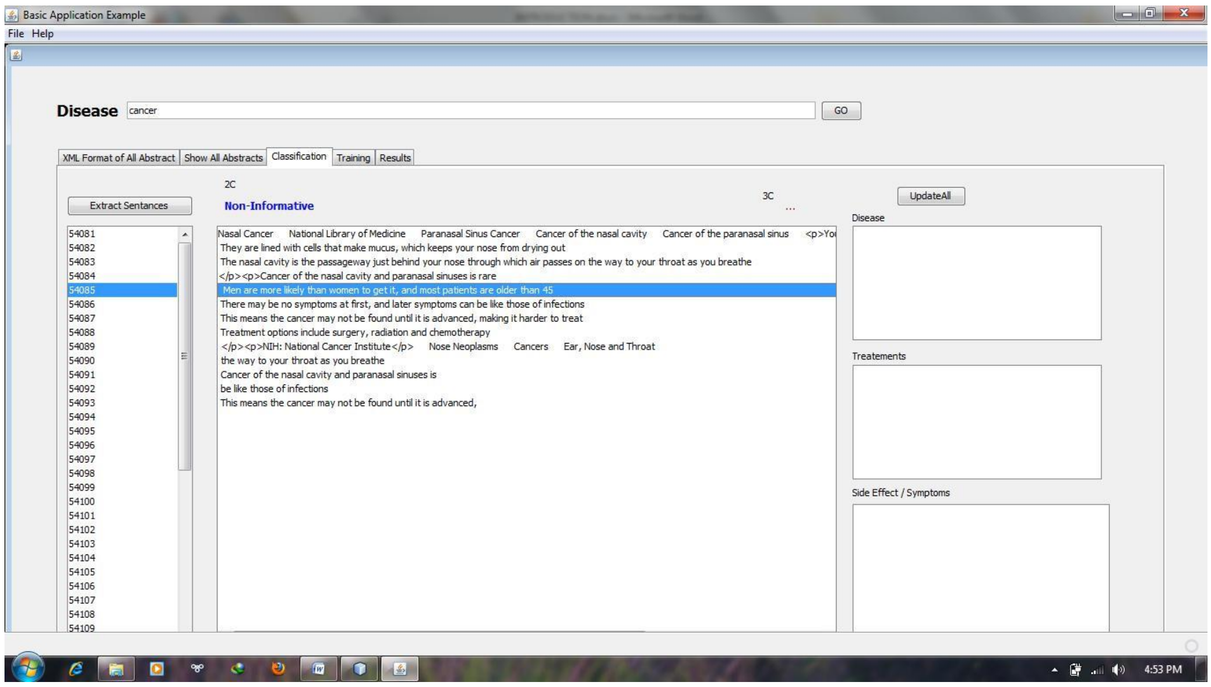
- A cough that doesn't go away and gets worse over time
- Constant chest pain
- Coughing up blood
- Shortness of breath, wheezing, or hoarseness
- Repeated problems with pneumonia or bronchitis
- Swelling of the neck and face
- Loss of appetite or weight loss
- Fatigue

There are many types of lung cancer. Each type of lung cancer grows and spreads in different ways and is treated differently. Treatment also depends on the stage, or how advanced it is. Treatment may include chemotherapy, radiation and surgery.

NIH: National Cancer Institute
Lung Neoplasms Cancers Lungs and Breathing Lung cancer is one of the most common cancers in the world. It is a leading cause of cancer death in men and women in the United States. Cigarette smoking causes most lung cancers. The more cigarettes you smoke per day and



```
Nasal Cancer    National Library of Medicine    Paranasal Sinus Cancer    Cancer of the nasal cavity    Cancer of the paranasal sinus    <p>Yo
They are lined with cells that make mucus, which keeps your nose from drying out
The nasal cavity is the passageway just behind your nose through which air passes on the way to your throat as you breathe
</p><p>Cancer of the nasal cavity and paranasal sinuses is rare
 Men are more likely than women to get it, and most patients are older than 45
There may be no symptoms at first, and later symptoms can be like those of infections
This means the cancer may not be found until it is advanced, making it harder to treat
Treatment options include surgery, radiation and chemotherapy
</p><p>NIH: National Cancer Institute</p>    Nose Neoplasms    Cancers    Ear, Nose and Throat
the way to your throat as you breathe
Cancer of the nasal cavity and paranasal sinuses is
be like those of infections
This means the cancer may not be found until it is advanced,
```

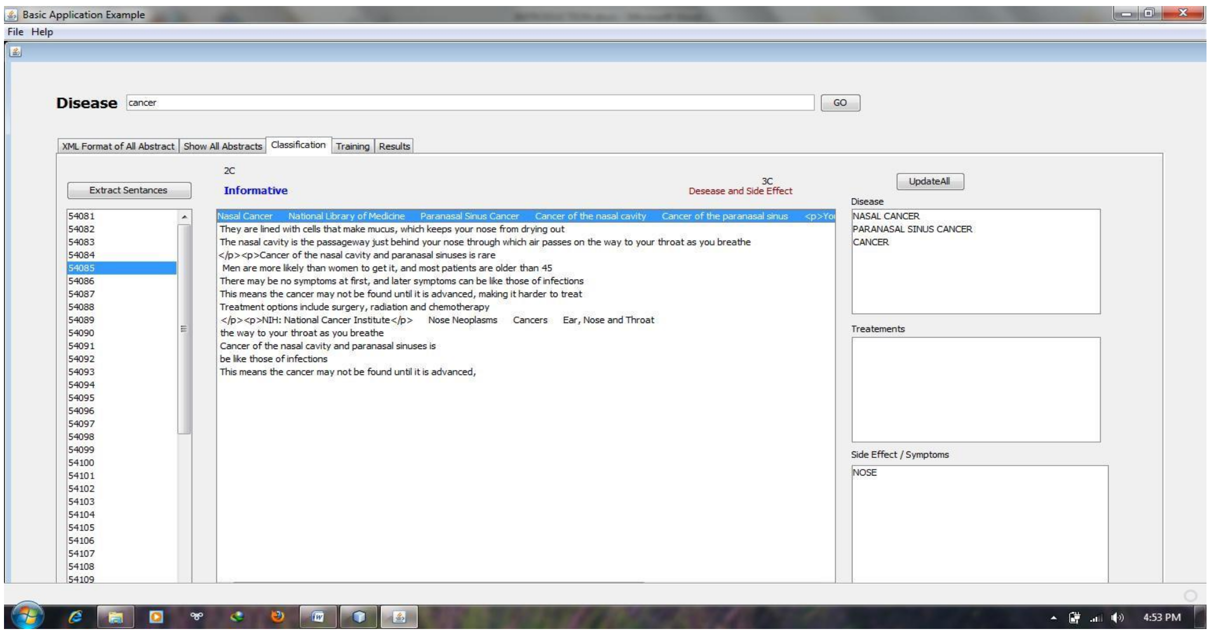


Disease: cancer

Non-Informative

```
54081
54082
54083
54084
54085
54086
54087
54088
54089
54090
54091
54092
54093
54094
54095
54096
54097
54098
54099
54100
54101
54102
54103
54104
54105
54106
54107
54108
54109
```

```
Nasal Cancer    National Library of Medicine    Paranasal Sinus Cancer    Cancer of the nasal cavity    Cancer of the paranasal sinus    <p>Yo
They are lined with cells that make mucus, which keeps your nose from drying out
The nasal cavity is the passageway just behind your nose through which air passes on the way to your throat as you breathe
</p><p>Cancer of the nasal cavity and paranasal sinuses is rare
 Men are more likely than women to get it, and most patients are older than 45
There may be no symptoms at first, and later symptoms can be like those of infections
This means the cancer may not be found until it is advanced, making it harder to treat
Treatment options include surgery, radiation and chemotherapy
</p><p>NIH: National Cancer Institute</p>    Nose Neoplasms    Cancers    Ear, Nose and Throat
the way to your throat as you breathe
Cancer of the nasal cavity and paranasal sinuses is
be like those of infections
This means the cancer may not be found until it is advanced,
```



Disease: cancer

Informative

Desease and Side Effect

```
54081
54082
54083
54084
54085
54086
54087
54088
54089
54090
54091
54092
54093
54094
54095
54096
54097
54098
54099
54100
54101
54102
54103
54104
54105
54106
54107
54108
54109
```

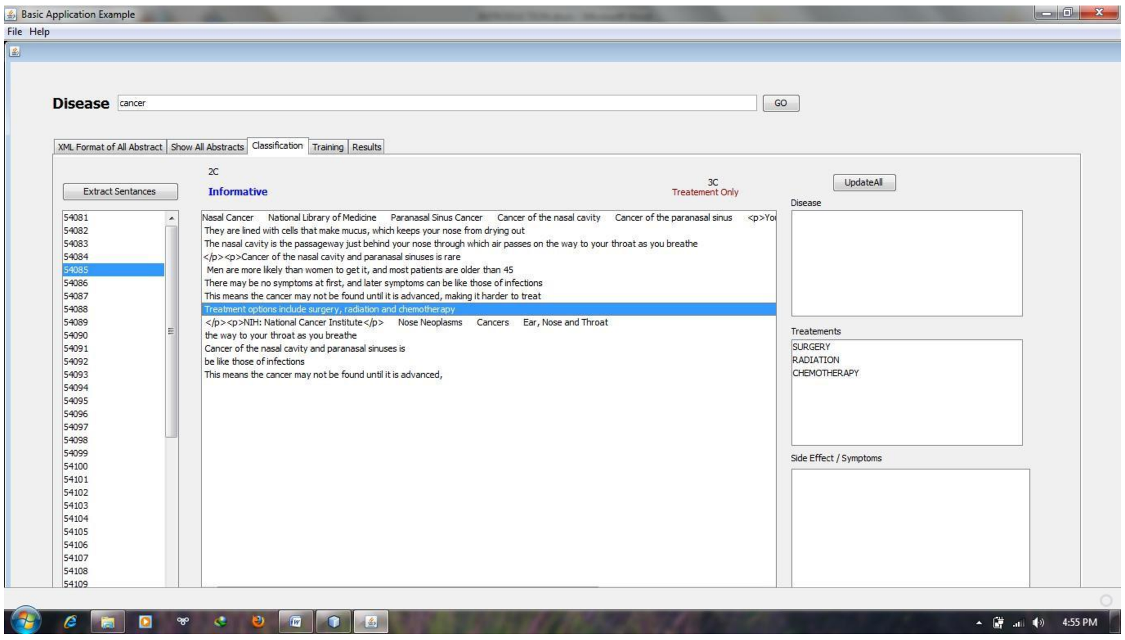
```
Nasal Cancer    National Library of Medicine    Paranasal Sinus Cancer    Cancer of the nasal cavity    Cancer of the paranasal sinus    <p>Yo
They are lined with cells that make mucus, which keeps your nose from drying out
The nasal cavity is the passageway just behind your nose through which air passes on the way to your throat as you breathe
</p><p>Cancer of the nasal cavity and paranasal sinuses is rare
 Men are more likely than women to get it, and most patients are older than 45
There may be no symptoms at first, and later symptoms can be like those of infections
This means the cancer may not be found until it is advanced, making it harder to treat
Treatment options include surgery, radiation and chemotherapy
</p><p>NIH: National Cancer Institute</p>    Nose Neoplasms    Cancers    Ear, Nose and Throat
the way to your throat as you breathe
Cancer of the nasal cavity and paranasal sinuses is
be like those of infections
This means the cancer may not be found until it is advanced,
```

Disease:

```
NASAL CANCER
PARANASAL SINUS CANCER
CANCER
```

Side Effect / Symptoms:

```
NOSE
```

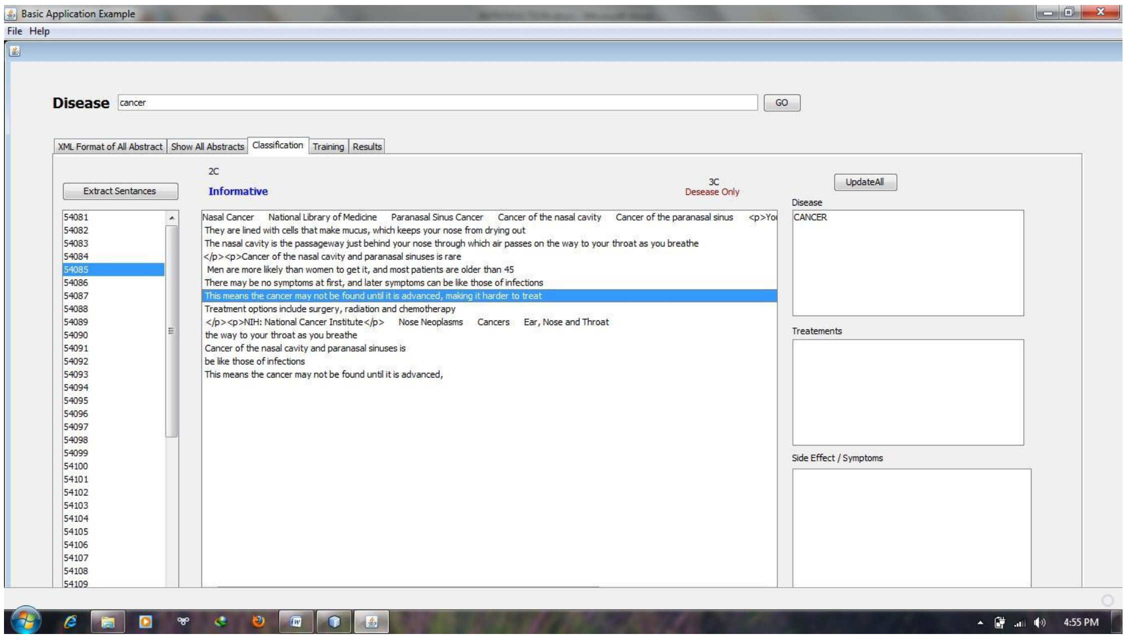


cancer

54081
54082
54083
54084
54085
54086
54087
54088
54089
54090
54091
54092
54093
54094
54095
54096
54097
54098
54099
54100
54101
54102
54103
54104
54105
54106
54107
54108
54109

```
Nasal Cancer    National Library of Medicine    Paranasal Sinus Cancer    Cancer of the nasal cavity    Cancer of the paranasal sinus    <p>Yo
 They are lined with cells that make mucus, which keeps your nose from drying out
 The nasal cavity is the passageway just behind your nose through which air passes on the way to your throat as you breathe
</p><p>Cancer of the nasal cavity and paranasal sinuses is rare
  Men are more likely than women to get it, and most patients are older than 45
 There may be no symptoms at first, and later symptoms can be like those of infections
 This means the cancer may not be found until it is advanced, making it harder to treat
 Treatment options include surgery, radiation and chemotherapy
 </p><p>NIH: National Cancer Institute</p>    Nose Neoplasms    Cancers    Ear, Nose and Throat
 the way to your throat as you breathe
 Cancer of the nasal cavity and paranasal sinuses is
 be like those of infections
 This means the cancer may not be found until it is advanced,
```

SURGERY
RADIATION
CHEMOTHERAPY

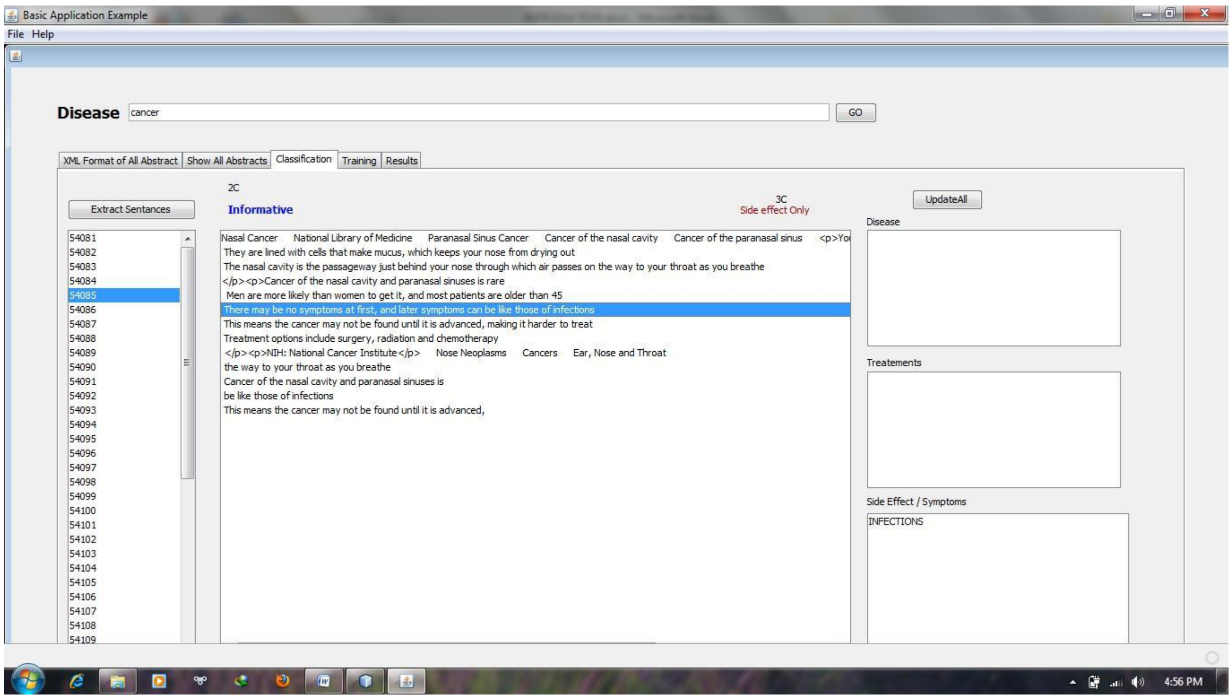


cancer

54081
54082
54083
54084
54085
54086
54087
54088
54089
54090
54091
54092
54093
54094
54095
54096
54097
54098
54099
54100
54101
54102
54103
54104
54105
54106
54107
54108
54109

```
Nasal Cancer    National Library of Medicine    Paranasal Sinus Cancer    Cancer of the nasal cavity    Cancer of the paranasal sinus    <p>Yo
 They are lined with cells that make mucus, which keeps your nose from drying out
 The nasal cavity is the passageway just behind your nose through which air passes on the way to your throat as you breathe
</p><p>Cancer of the nasal cavity and paranasal sinuses is rare
  Men are more likely than women to get it, and most patients are older than 45
 There may be no symptoms at first, and later symptoms can be like those of infections
 This means the cancer may not be found until it is advanced, making it harder to treat
 Treatment options include surgery, radiation and chemotherapy
 </p><p>NIH: National Cancer Institute</p>    Nose Neoplasms    Cancers    Ear, Nose and Throat
 the way to your throat as you breathe
 Cancer of the nasal cavity and paranasal sinuses is
 be like those of infections
 This means the cancer may not be found until it is advanced,
```

CANCER



Disease: cancer

```
54081
54082
54083
54084
54085
54086
54087
54088
54089
54090
54091
54092
54093
54094
54095
54096
54097
54098
54099
54100
54101
54102
54103
54104
54105
54106
54107
54108
54109
```

2C
**Informative**

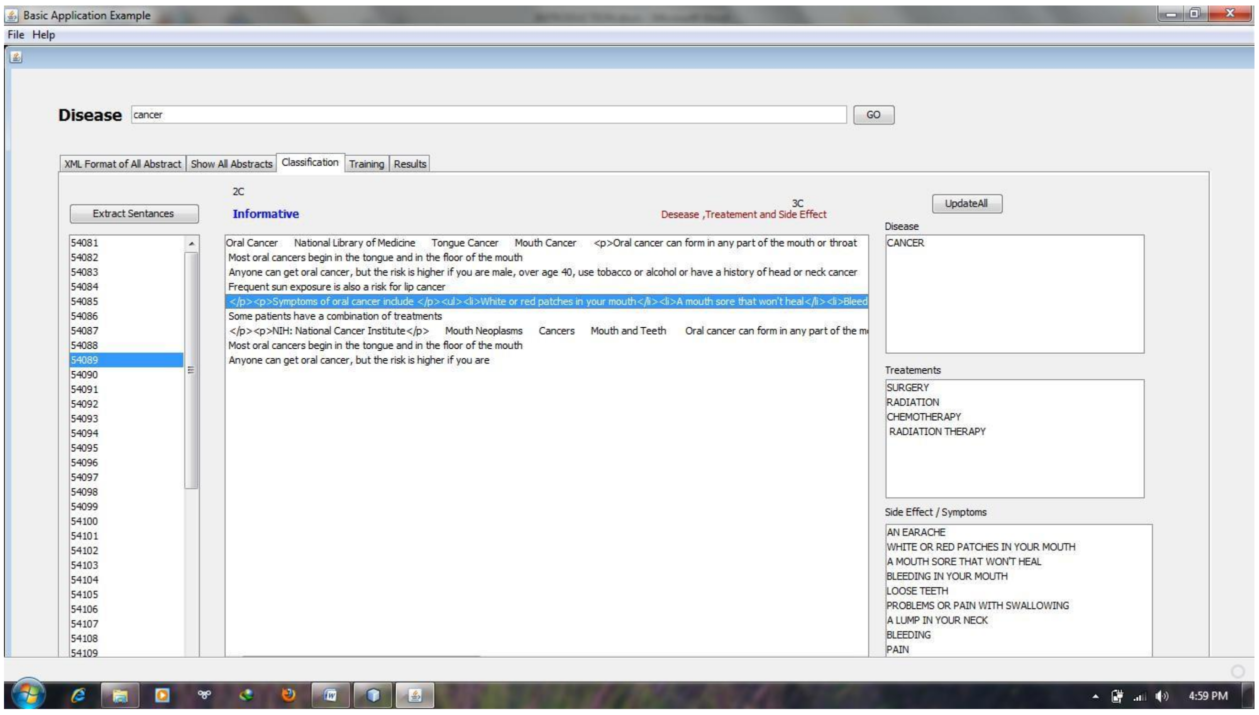
3C
Side effect Only

```
Nasal Cancer     National Library of Medicine     Paranasal Sinus Cancer     Cancer of the nasal cavity     Cancer of the paranasal sinus     <p>Yo
They are lined with cells that make mucus, which keeps your nose from drying out
The nasal cavity is the passageway just behind your nose through which air passes on the way to your throat as you breathe
</p><p>Cancer of the nasal cavity and paranasal sinuses is rare
 Men are more likely than women to get it, and most patients are older than 45
There may be no symptoms at first, and later symptoms can be like those of infections
This means the cancer may not be found until it is advanced, making it harder to treat
Treatment options include surgery, radiation and chemotherapy
</p><p>NIH: National Cancer Institute</p>     Nose Neoplasms     Cancers     Ear, Nose and Throat
the way to your throat as you breathe
Cancer of the nasal cavity and paranasal sinuses is
be like those of infections
This means the cancer may not be found until it is advanced,
```

Disease

Treatements

Side Effect / Symptoms
INFECTIONS



Disease: cancer

```
54081
54082
54083
54084
54085
54086
54087
54088
54089
54090
54091
54092
54093
54094
54095
54096
54097
54098
54099
54100
54101
54102
54103
54104
54105
54106
54107
54108
54109
```

2C
**Informative**

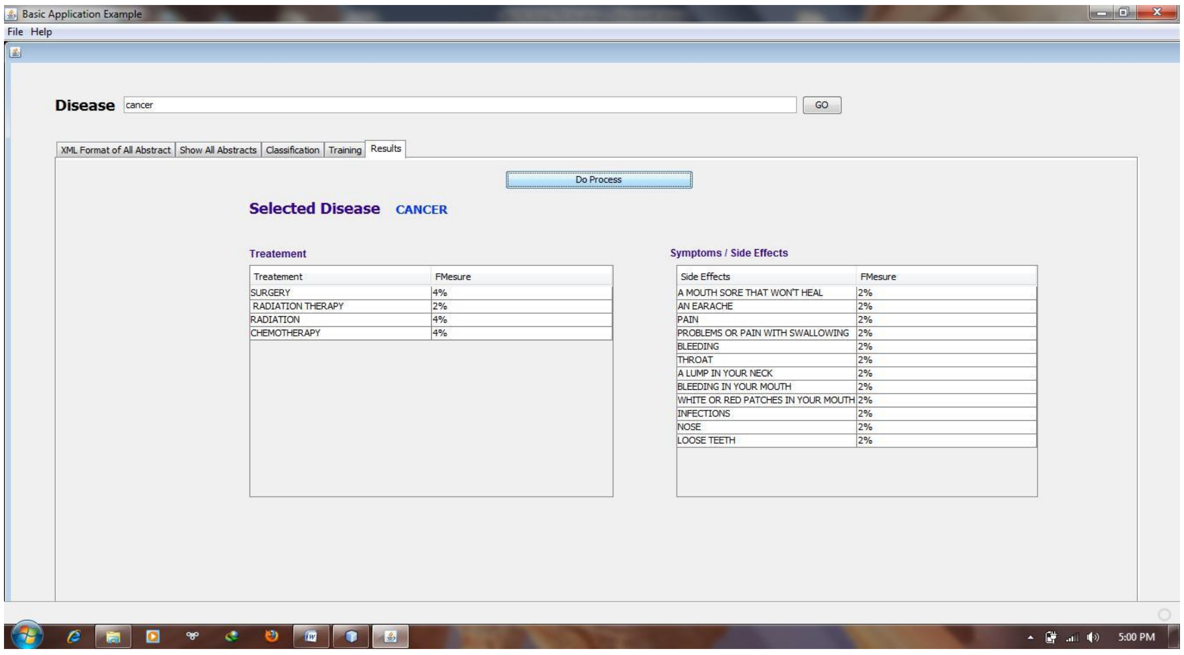
3C
Desease ,Treatement and Side Effect

```
Oral Cancer     National Library of Medicine     Tongue Cancer     Mouth Cancer     <p>Oral cancer can form in any part of the mouth or throat
Most oral cancers begin in the tongue and in the floor of the mouth
Anyone can get oral cancer, but the risk is higher if you are male, over age 40, use tobacco or alcohol or have a history of head or neck cancer
Frequent sun exposure is also a risk for lip cancer
</p><p>Symptoms of oral cancer include </p><ul><li>White or red patches in your mouth</li><li>A mouth sore that won't heal</li><li>Bleed
Some patients have a combination of treatments
</p><p>NIH: National Cancer Institute</p>     Mouth Neoplasms     Cancers     Mouth and Teeth     Oral cancer can form in any part of the m
Most oral cancers begin in the tongue and in the floor of the mouth
Anyone can get oral cancer, but the risk is higher if you are
```

Disease
CANCER

Treatements
SURGERY
RADIATION
CHEMOTHERAPY
RADIATION THERAPY

Side Effect / Symptoms
AN EARACHE
WHITE OR RED PATCHES IN YOUR MOUTH
A MOUTH SORE THAT WON'T HEAL
BLEEDING IN YOUR MOUTH
LOOSE TEETH
PROBLEMS OR PAIN WITH SWALLOWING
A LUMP IN YOUR NECK
BLEEDING
PAIN



Disease: cancer

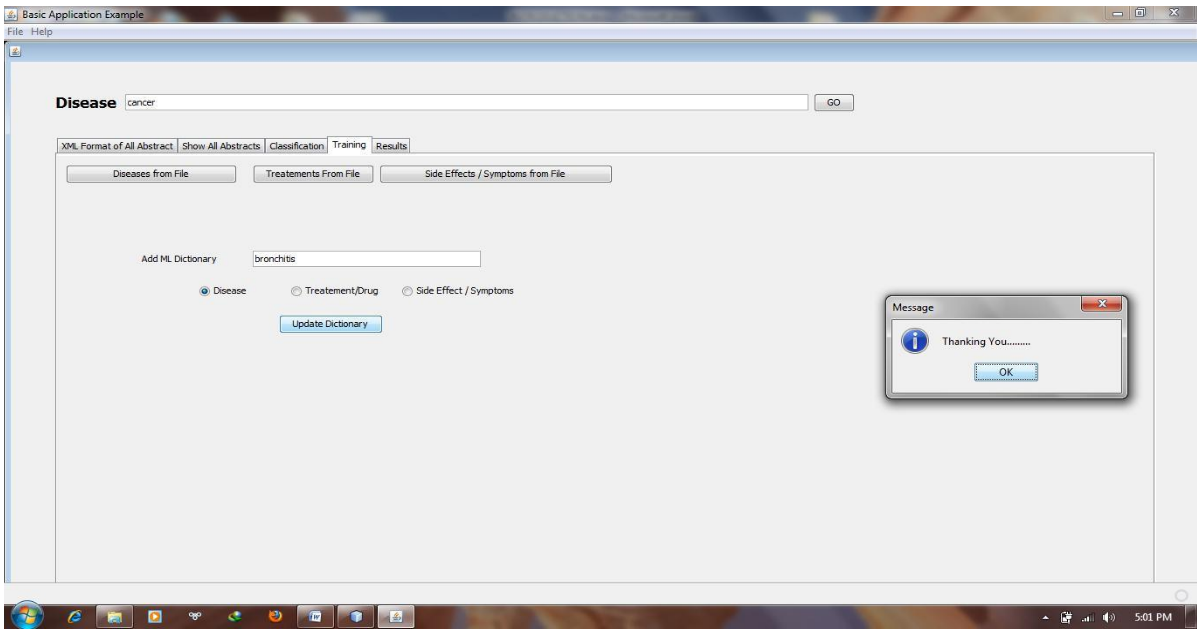
Selected Disease CANCER

Treatement

| Treatement | FMesure |
|---|---|
| SURGERY | 4% |
| RADIATION THERAPY | 2% |
| RADIATION | 4% |
| CHEMOTHERAPY | 4% |

Symptoms / Side Effects

| Side Effects | FMesure |
|---|---|
| A MOUTH SORE THAT WON'T HEAL | 2% |
| AN EARACHE | 2% |
| PAIN | 2% |
| PROBLEMS OR PAIN WITH SWALLOWING | 2% |
| BLEEDING | 2% |
| THROAT | 2% |
| A LUMP IN YOUR NECK | 2% |
| BLEEDING IN YOUR MOUTH | 2% |
| WHITE OR RED PATCHES IN YOUR MOUTH | 2% |
| INFECTIONS | 2% |
| NOSE | 2% |
| LOOSE TEETH | 2% |



Disease: cancer

Add ML Dictionary: bronchitis

[✓ Disease] Treatement/Drug Side Effect / Symptoms

Thanking You........

# 11.CONCLUSION AND FUTURE ENHANCEMENTS

## CONCLUSION

This project suggest that domain-specific knowledge improves the results. Probabilistic models are stable and reliable for tasks performed on short texts in the medical domain. The representation techniques influence the results of the ML algorithms, but more informative representations are the ones that consistently obtain the best results.

## FUTURE ENHANCEMENT

As future work, we would like to extend the experimental methodology when the first setting is applied for the second task, to use additional sources of information as representation techniques, and to focus more on ways to integrate the research discoveries in a framework to be deployed to consumers. In addition to more methodological settings in which we try to find the potential value of other types of representations, we would like to focus on source data that comes from the web. Identifying and classifying medical-related information on the web is a challenge that can bring valuable information.

# 12. BIBLIOGRAPHY

[1] R. Bunescu and R. Mooney, “A Shortest Path Dependency Kernelfor Relation Extraction,”Proc. Conf. Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP), pp. 724-731, 2005.

[2] R. Bunescu, R. Mooney, Y. Weiss, B. Scho¨ lkopf, and J. Platt,“Subsequence Kernels for Relation Extraction,” Advances in Neural Information Processing Systems, vol. 18, pp. 171-178, 2006.

[3] A.M. Cohen and W.R. Hersh, and R.T. Bhupatiraju, “Feature Generation, Feature Selection, Classifiers, and Conceptual Drift for Biomedical Document Triage,” Proc. 13th Text Retrieval Conf.(TREC), 2004.

[4] M. Craven, “Learning to Extract Relations from Medline,” Proc.Assoc. for the Advancement of Artificial Intelligence, 1999.

[5] I. Donaldson et al., “PreBIND and Textomy: Mining the Biomedical Literature for Protein- Protein Interactions Using a Support Vector Machine,” BMC Bioinformatics, vol. 4, 2003.

[6] C. Friedman, P. Kra, H. Yu, M. Krauthammer, and A. Rzhetsky,“GENIES: A Natural Language Processing System for the Extraction of Molecular Pathways from Journal Articles,” Bioinformatics,vol. 17, pp. S74-S82, 2001.

[7] O. Frunza and D. Inkpen, “Textual Information in Predicting Functional Properties of the Genes,” Proc. Workshop Current Trendsin Biomedical Natural Language Processing (BioNLP) in conjunction with Assoc. for Computational Linguistics (ACL ’08), 2008.

[8] R. Gaizauskas, G. Demetriou, P.J. Artymiuk, and P. Willett,“Protein Structures and Information Extraction from BiologicalTexts: The PASTA System,” Bioinformatics, vol. 19, no. 1, pp. 135-143, 2003.

[9] C. Giuliano, L. Alberto, and R. Lorenza, “Exploiting Shallow Linguistic Information for Relation Extraction from Biomedical Literature,” Proc. 11th Conf. European Chapter of the Assoc. for Computational Linguistics, 2006.

[10] J. Ginsberg, H. Mohebbi Matthew, S.P. Rajan, B. Lynnette, S.S.Mark, and L. Brilliant, "Detecting Influenza Epidemics Using Search Engine Query Data," Nature, vol. 457, pp. 1012- 1014, Feb.2009.

[11] M. Goadrich, L. Oliphant, and J. Shavlik, "Learning Ensembles of First-Order Clauses for Recall-Precision Curves: A Case Study in Biomedical Information Extraction," Proc. 14th Int'l Conf. InductiveLogic Programming, 2004.

[12] L. Hunter and K.B. Cohen, "Biomedical Language Processing:What's beyond PubMed?" Molecular Cell, vol. 21-5, pp. 589-594,2006.

[13] L. Hunter, Z. Lu, J. Firby, W.A. Baumgartner Jr., H.L. JohnsonP.V. Ogren, and K.B. Cohen, "OpenDMAP: An Open Source,Ontology-Driven Concept Analysis Engine, with Applications toCapturing Knowledge Regarding Protein Transport, Protein Interactions and Cell-Type- Specific Gene Expression," BMCBioinformatics, vol. 9, article no. 78, Jan. 2008.

[14] T.K. Jenssen, A. Laegreid, J. Komorowski, and E. Hovig, "ALiterature Network of Human Genes for High-ThroughputAnalysis of Gene Expression," Nature Genetics, vol. 28, no. 1,pp. 21-28, 2001.

[15] R. Kohavi and F. Provost, "Glossary of Terms," Machine Learning,Editorial for the Special Issue on Applications of MachineLearning and the Knowledge Discovery Process, vol. 30,

pp. 271-274, 1998.

[16] G. Leroy, H.C. Chen, and J.D. Martinez, "A Shallow Parser Based on Closed-Class Words to Capture Relations in Biomedica lText," J. Biomedical Informatics, vol. 36, no. 3, pp. 145-158,

2003.

[17] J. Li, Z. Zhang, X. Li, and H. Chen, "Kernel-Based Learning forBiomedical Relation Extraction," J. Am. Soc. Information Science andTechnology, vol. 59, no. 5, pp. 756-769, 2008.

[18] T. Mitsumori, M. Murata, Y. Fukuda, K. Doi, and H. Doi,"Extracting Protein-Protein Interaction Information from BiomedicalText with SVM," IEICE Trans. Information and Systems,vol. E89D, no. 8, pp. 2464-2466, 2006.

[19] M. Yusuke, S. Kenji, S. Rune, M. Takuya, and T. Jun'ichi,"Evaluating Contributions of Natural Language Parsers to Protein-Protein Interaction Extraction," Bioinformatics, vol. 25,

pp. 394-400, 2009.

[20] S. Novichkova, S. Egorov, and N. Daraselia, "MedScan, A Natural Language Processing Engine for MEDLINE Abstracts," Bioinformatics,vol. 19, no. 13, pp. 1699-1706, 2003.

[21] M. Ould Abdel Vetah, C. Ne´dellec, P. Bessie`res, F. Caropreso,A.-P.Manine, and S. Matwin, "Sentence Categorization inGenomics Bibliography: A Naive Bayes Approach," Actes de laJourne´eInformatiqueetTranscriptome, J.-F. Boulicaut andM. Gandrillon, eds., Mai 2003.

[22] J. Pustejovsky, J. Castan˜ o, J. Zhang, M. Kotecki, and B. Cochran,"Robust Relational Parsing over Biomedical Literature: Extracting Inhibit Relations," Proc. Pacific Symp. Biocomputing, vol. 7, pp. 362-373, 2002.

[23] S. Ray and M. Craven, "Representing Sentence Structure in Hidden Markov Models for Information Extraction," Proc. Int'lJoint Conf. Artificial Intelligence (IJCAI '01), 2001.

[24] T.C. Rindflesch, L. Tanabe, J.N. Weinstein, and L. Hunter,"EDGAR: Extraction of Drugs, Genes, and Relations from theBiomedical Literature," Proc. Pacific Symp. Biocomputing, vol.5,

pp. 514-525, 2000.

[25] B. Rosario and M.A. Hearst, "Semantic Relations in BioscienceText," Proc. 42nd Ann. Meeting on Assoc. for Computational Linguistics, vol. 430, 2004.

[26] P. Srinivasan and T. Rindflesch, "Exploring Text Mining from Medline," Proc. Am. Medical Informatics Assoc. (AMIA) Symp.,2002.

[27] B.J. Stapley and G. Benoit, "Bibliometrics: Information Retrieval Visualization from Co- Occurrences of Gene Names in MEDLINEAbstracts," Proc. Pacific Symp. Biocomputing, vol. 5, pp. 526-537,2000.

[28] J. Thomas, D. Milward, C. Ouzounis, S. Pulman, and M. Carroll,“Automatic Extraction of Protein Interations from ScientificAbstracts,” Proc. Pacific Symp. Biocomputing, vol. 5, pp. 538- 549,2000.

[29] A.Yakushiji, Y. Tateisi, Y. Miyao, and J. Tsujii, “Event Extraction from Biomedical Papers Using a Full Parser,” Proc. Pacific Symp.Biocomputing, vol. 6, pp. 408-419, 2001.