Automated Disease Prediction System (ADPS): A User Input-Based Reliable Architecture for Disease Prediction

Submitted by:

Md. Tahmid Hossain Student No. 114418

Md. Tahmid Rahman Laskar Student No. 114426

Supervisor:
Dr. Abu Raihan Mostofa Kamal
Associate Professor,
Department of CSE,
IUT.

Co Supervisor:
Nafiul Rashid
Lecturer,
Department of CSE,
IUT.

Computer Science and Engineering
Islamic University of Technology
October 30, 2015.

Table of Contents

Chapter 1
INTRODUCTION

1.1 Background

Number of internet users is growing exponentially over the years. In a national survey conducted by the Pew Internet Project [2] found that 72% of Internet users in the United States, have gone online in search of health information. People post their health related queries (such as asking about what kind of disease that they might be suffering from) on various healthcare forums. There are other group of people who leave their responses to those posts with predictions of possible diseases. However, these predictions may not be always accurate, and also there is no assurance that users will always get a reply on their post. Moreover, some posts are fabricated or made up which can drive the patient in a wrong direction. It is worth noting that a huge number of users on these forums hold fake identities. According to a survey conducted by CNN [4], it is found that 25% users lie on social networking sites. Therefore, reliability is a big issue here.

Substantial amount of research work on automated disease prediction is going on in recent years. It can be classified in two major categories: One is disease prediction based on specialized/clinical text source and another is disease prediction based on unspecialized text source. Bulk of the research work focused on predicting diseases automatically from specialized text sources like clinical reports [1]. However, predicting disease based on user (patient) input is a complete different ball game ([3], [13] and [16]). Generally, people express their symptoms in non-technical or natural terms which adds complexity in predicting diseases. Diagnosis decision support systems provide users with probable diagnosis based on user input.

1.2 Automated Disease Prediction System

Diagnosis decision support system is a computer based program that helps users in diagnosing diseases. Many research organization and companies developed diagnosis support systems using different technologies and provide various levels of functionalities. User enters the symptoms and the system processes the input and comes up with probable diagnosis. Diagnosis system that allows user to enter symptoms in natural language text are not fully reliable and often comes up with false diagnosis which makes users confused to determine with which disease they might be suffering from.

1.3 Importance of improving diagnosis decision making

Diagnosis is the first and most important decision made about the patient. It determines all subsequent treatment and determines the course of each patient encounter. How well this decision is made, therefore, is one of the most significant determinants of healthcare quality and efficiency. The following are some of the areas where the speed and accuracy of diagnosis have a key impact and where the use of diagnosis decision aids could help effect improvements:

Referrals from primary care to specialists: Research shows that 30-50% of referrals from primary care to specialists are inappropriate leading to delays in diagnosis, patient dissatisfaction and lengthy waits at specialist clinics.
Test ordering: Surveys and anecdotal evidence put the level of unnecessary and defensive test ordering at 40%. This is extremely costly and leads patients to unnecessary clinical risk through invasive procedures and radiation exposure.
Medical malpractice: Misdiagnosis accounts for 30-40% of all malpractice claims and about 2/3 of all claims in primary care. Additionally, diagnostic errors are frequently the leading or second leading cause of malpractice claims in the United States, accounting for twice as many alleged and settled claims as medication errors.
Patient satisfaction: Because patient satisfaction will soon account for 30% of Medicare payments, many hospitals are investing in typical customer service initiatives used for years in other industries. However, in many cases these are viewed as gimmicks by patients and will not make up for poor quality of care. A survey of patients' concerns showed that their top concern when visiting their primary care physician is diagnosis and in hospitals it is their 2nd most important concern.
Employee skills: Healthcare is a knowledge-intensive industry and a key issue underpinning an institution's success is the clinical skills of all its clinicians. One way of boosting skills across the board is to provide tools that increase clinical skills. Although diagnosis is traditionally seen as the preserve of the physicians, it is the nurses who are caring for the patient most of the time and improving their diagnosis skills can lead to an improved level of patient safety and quality of care.

1.4 History of diagnosis decision support systems

Most of the basic concepts related to clinical diagnosis support systems were formulated before or in early 1970. In 1979 review of reasoning strategies by Shortliffe, Buchanan, and Feigenbaum identified the following methods:

1. Clinical algorithms.
2. Clinical databanks that include analytical functions.
3. Mathematical patho-physiological models.
4. Pattern recognition systems.
5. Bayesian statistical systems.

6. Decision-analytical systems.
7. Symbolic reasoning.

In 1959, Ledley and Lusted published a paper that clinicians have imperfect knowledge of how they solve clinical diagnostic problems, and they published the principles underlying work on Bayesian and clinical diagnostic support systems that has been followed over new few decades. Their detailed logic and probabilistic reasoning was most important parts of the human diagnostic reasoning. Bayes' rule can be applied to larger areas and in 1960-1961 Warner and colleagues developed one first medical application systems based on Bayes' rule.

Gorry and Barnett in 1968 developed a model for sequential Bayesian diagnosis, Dombal and colleagues developed first practical Bayesian system to diagnose acute pain abdomen and that is one of the first clinical diagnostic support system that used at many medical centers. Bayesian methods gained popularity and many other developers developed diagnostic related systems using Bayesian logic.

In late 1950s, Lipkin, Hardy, Engle, and their colleagues [25] developed a first heuristics based diagnostic application called HEME to diagnose hematological disorders. HEME program heuristically matches stored disease data to lexical description of patient's clinical findings. Later CONSIDER system was developed by Lindberg and RECONSIDER program was developed by Blois and his colleagues using heuristic lexical matching techniques. Weiss and Kulikowski developed EXPERT system shell and it has been used in systems that utilize criteria tables for example AI/Rheun mainly developed for diagnosis of rheumatological disorders.

In 1968 Gorry published general principles for expert system approach to clinical diagnosis systems, based on the Gorry's principles clinical diagnostic systems were developed in 1970 and 1980. Gorry principles demonstrate many clinical diagnostic systems developed by various development groups, systems including PIP (the Present Illness Program), developed by Pauker et al, MEDITEL for adult illnesses developed by Waxman and Worley from its predecessor pediatric version, Internist-I developed by Pople and Myers Miller in University of Pittsburgh, QMR developed by Miller, Masarie, and Myers, and DXplain developed by Barnett and colleagues, Iliad developed by Warner and colleagues, and many other systems developed by diverse group.

Shortliffe introduced the rule based expert system to develop medical applications; many rule based clinical diagnosis support systems were developed over the years but rule based expert systems are applied only in small area of the domain because of its complexity in maintaining thousands of predefined rules. The philosophy of clinical diagnosis software systems development has been changed with advent and proliferation of the personal Clinical Diagnosis Support Systems 22 computers. Developers developed systems that take advantage of strengths of user knowledge and the system capabilities. The goal of the developers was to improve performance of the user and machine capabilities.

In 1980s and 1990s several advanced techniques were developed to existing clinical diagnosis software systems and models and improvements were made with adding more mathematical rigor to the models. However mathematical approaches have one downside that is they are dependent

on the quality of the data. Many systems were developed based on fuzzy set theory and Bayesian belief networks logic to overcome limitations of heuristic approaches and the old models.

With advent of artificial neural networks and artificial intelligence, developers and researchers are taking completely new approach to develop clinical diagnosis decision support systems. Even though simple neural network may be similar to Bayesian probabilities logic but in general neural networks technology is very complex requires lot of patient's data to train the neural network. Use of artificial patient data to train the neural network may not be realistic and may affect its performance on real patient's data.

Some important methodologies & technologies for clinical decision support are Information retrieval, evaluation of logical conditions, probabilistic and data-driven classification or prediction, heuristic modeling and expert systems, calculations, algorithms, and multistep processes and associative groupings of elements.

1.5 Available diagnosis decision making aids

With the nature of the diagnostic process, technology advances have long been seen as potentially useful tools to help support the clinician. Initial attempts in the 1960's were focused on the improvement in diagnosis of one specific problem, such as abdominal pain. Although these showed that clinicians did a better job when using them, the tools were time-consuming and proved to be impractical for use in a busy clinical setting, so they were never adopted. Another factor was that the intended users were specialists and had less need of the tools.

The 1970's and 80's brought the first general diagnostic tools such as *DxPlain, QMR, Diagnosis Pro* and *Iliad*. These tools were also not widely adopted, primarily due to the time taken to use.

Although *DxPlain* and *Diagnosis Pro* are still available, *QMR* and *Iliad* have all but faded away. These systems were highly developed, but were limited by the technology available when they were launched. The tools are "rules-based systems," which means that each symptom is associated with a particular disease with an assigned probability. These systems work satisfactorily on a small scale, but become difficult to manage on a large scale as each symptom or diagnosis needs to be kept up to date. The rigid nature of a rules-based system also means that the user can only enter a feature that is in the system's database. A by-product of this problem is that it makes it more difficult to fully integrate these systems into electronic medical records.

Problem Knowledge Couplers was started by the father of the problem-orientated medical note, Larry Weed. This system is more focused on providing a structure for the initial history taking to ensure that the right questions are asked. The main deployment of PKC has been within the Department of Defense (DOD). The system's use within the DOD was evaluated after two years of use but the study concluded, noting: "This study provides no strong evidence to support the utility of this decision-support tool, but it demonstrates the value of rigorous evaluation of decision-support information technology."

Isabel marked the new generation of diagnostic tools and was first introduced in 2001. Isabel uses a statistical natural language processing (SNLP) engine applied to a database of disease presentations rather than a rules based model.

IBM Watson has more recently entered the medical diagnosis field, seeking to adapt its Jeopardy winning system into a tool for diagnosis and treatment. IBM expects to have the first pilot version ready in 2014. Watson aims to use both SNLP and NLP applied to a broad base of 200mn documents from textbooks through blogs.

VisualDx is another system but is based on digital images and allows clinicians to build a visual differential diagnosis based on actual patient findings.

Google.com is also commonly used as a diagnosis aid. In 2006, the BMJ ran a study entitled "Googling for a diagnosis –Use of Google as a diagnostic aid: internet based study".6 The results showed that Google included the final diagnosis in 58% of cases but only when "statistically improbable phrases" were entered and three possible diagnoses were pre-selected from Google's list of documents by 2 specialists. It should be noted that Isabel and DxPlain found the final diagnosis under more realistic test conditions in the 90% range.

1.6 Diagnosis errors and solutions

Literature on diagnosis error abounds (see Appendix to read more), showing that the causes of delays and errors in diagnosis are many, which means that there is no single intervention that can solve the problem. Some causes are system related, such as test results being misplaced or not received by the physician, and therefore not acted on or communicated to the patient. It is hoped that the introduction of electronic medical records and other technologies, like personal health records and patient based tools, will help reduce the system related causes. However, the majority of causes are related to how physicians think and the process of working up a patient's diagnosis. There are many intrinsic attributes to us as human beings that contribute to causing diagnosis related errors.

Premature Closure: The more common causes of diagnosis error are due to how a doctor thinks. There is now a large body of work describing the many biases that we, as human beings and not just clinicians, are prone to. The research now lists over a 100 different biases but the main types that cause the errors in diagnosis are the availability ones. In a time-constrained industry this is to be expected. As Dr. Mark Graber described his landmark paper, "Diagnostic error in internal medicine," a classic cause is "premature closure," where the clinician decides on a diagnosis very quickly, but then fails to consider other reasonable possibilities until it is too late. In any analysis of cases where the diagnosis was delayed or missed, premature closure has been the most common contributing bias.

Cognitive De-biasing: One of the proposed solutions to this cognitive problem is termed "cognitive de-biasing" and involves clinicians being made aware of these issues as part of their

medical training. This solution will help, but in order to be sustainable, it needs to be accompanied by the routine use of tools to help at the point of care.

Differential Diagnosis: Another solution commonly proposed is actually very old and is the routine construction of a comprehensive, differential diagnosis. Olga Kostopoulou has carried out a number studies looking at the predictors of diagnostic accuracy, including "Missing celiac disease in family medicine: the importance of hypothesis generation"4 and "Diagnosis of difficult cases in primary care."5 In the research, Kostopulou found that the most significant factor is having a good differential diagnosis that includes what turns out to be the correct diagnosis.

Although the construction and use of a comprehensive differential diagnosis has been taught for over 100 years, it is not used routinely in medicine. One of the main reasons for this is the time needed to construct one. Due to a lack of time in the ED or primary care, for example, many clinicians rely on their memory to construct a differential. However, with a universe of diagnoses in primary care being only 200-300 compared to a total universe of about 12,000 diseases it is obvious that, on occasions, a clinician will simply not think of a diagnosis either because he did not remember it or never knew it in the first place.

If there is a diagnostic doubt, the clinician then typically has to consult with colleagues, read textbooks or research online in order to investigate further. With medical textbooks and online reference resources, it is very difficult to search for something when one does not know what to look for. A search for "toxic shock," for example, will provide huge amounts of information; but, if you are unsure and just know that the patient has ankle pain, ankle edema, diarrhea and fever, then the traditional reference resources are not very helpful in connecting and making sense of all of these signs and symptoms. In these more unusual or complex clinical presentations, diagnostic decision aids can be particularly helpful, as they are designed to produce a list of likely diagnoses for a given set of signs and symptoms.

Their job is to get the clinician thinking about a disease that he had not thought about previously. Instead of taking several hours, days or even years in some cases to suggest the right diagnosis using the traditional methods, the diagnosis decision aids work in seconds. These tools buy the time that the clinician needs to think.

Chapter 2
LITERATURE REVIEW

2.1 Related Work

In this chapter, we will discuss about the related works and detailed description of current diagnosis decision support systems.

2.1.1 Sentiment Analysis and Opinion Mining

Wang et al. used Clinical reports as data source provided by Newyork presbyterian Hospital.They used MedLee natural language processing system which parses necessary information from these reports into Extended Markup language (XML). They considered a disease and symptom to co-occur if they appeared in the same case report. All co-occurrence tables that had a frequency of less than 2 were excluded because they were very unlikely to yield meaningful results. They used statistical analysis to predict disease symptom association.

Ontology-Based Text Mining for Predicting Disease Outbreaks focused on detecting and predicting infectious disease outbreaks and bioterrorism. They mainly mined trending online news reports and vital social media contents for prediction. They used BioCaster Multilingual Ontology to mine text based data for disease prediction. It has a great scope of work space as it supports almost all the major languages. There are millions of entities on these knowledge bases. But detection and disambiguation of entities in natural language text, discovering newly emerging knowledge sources, linkage between many knowledge and data sources are challenging.

Research has been done that provides a lightweight method for using discourse relations for polarity detection of informal/formal opinion of any user on twitter. This method is targeted towards the web-based applications that deal with noisy, unstructured text, like the tweets, and cannot afford to use heavy linguistic resources like parsing due to frequent failure of the parsers to handle noisy data.

Most of the works in micro-blogs like Twitter, use a bag-of-words model that ignores the discourse particles like but, since, although etc. Detecting positivity or negativity of a certain opinion from a user sometimes becomes more vital than the content itself. Like in feedback blog of a commercial brands product, what is said by any user is important but in a nutshell what mindset the users have (positive feedback of the user or negative feedback of the user) is of great significance too. This research gives an insight on how the discourse relations like the connectives and conditionals can be used to incorporate discourse information in any bag-of-words model. Strong modals, weak

modals, conjunctions, negation, Inference and such other linguistic discourse features are used in such a way that polarity detection or sentiment analysis becomes easier. In this paper table of such clauses are maintained with corresponding inferential meaning.

Researches have been done that proposes a method for automatically establishing the credibility of user-generated medical statements in an online blog/website and the trustworthiness of their authors by exploiting linguistic features and distant supervision.It also focuses on drug side effect detection from comments.

To achieve the desired goal a probabilistic model has been designed that jointly learns user trustworthiness, statement credibility, and language objectivity.

User trustworthiness is measure based on his/her past credentials, up votes provided by other users and the feedback of the attention seeker in the first place. Moreover, trust factor of a particular user is analyzed with respect to interaction among him and other highly trustworthy users. e.g. If more than one highly trusted users defer in opinion with another user in a certain case, the lone person arguing will have his credibility and trustworthiness reduced.

Statement credibility is measured mainly by the linguistic features. For example, weak modals indicate vague idea of the user over the described problem whereas strong modals infer that the statement is provided by a person holding strong sense of the real problem.

This research provides a methodology which is applied to unveil rare/unknown side-effects of medical drugs—this being one of the areas where large scale non-expert data has the potential to endorse expert data.

As many diseases are interrelated and a drug or combination of different drugs can have manifold side effects, a person with such experience can really come up with the difficulties he faced and knowledge can be deduced from such a statement.

Web Contains significant amount of untruthful information. So it is necessary to have good tools to determine truthfulness of information. In this research a two steps method is used to determine if a statement is true or not, and if it is false then truthful statement most related to the given statement is searched. On the first step for a given statement, alternative statements related to it are generated and it is assured that one of those statements is true. Then the truthful statement among all the statements is found out. An example can be “Obama is a Muslim “. If it is searched on web which is false, then some alternative truthful statements would be generated and among those statements the truthful statement will be “Obama is a Christian“.

2.1.2 Knowledge Harvesting

Many researches have been done on this knowledge harvesting and knowledge linking mechanism. Like an entity on knowledge bases can have multiple classes. It is important to find the correct entity of a specific class or vice-versa. Research has been done to harvest knowledge from those sources which is robust to noise. As no knowledge bases are complete and there can be many entities in those knowledge bases which are uncovered. Open domain extraction is used to overcome this issue. There are huge amount of data available on web about diseases, drugs,symptoms etc. Also there are many popular health communities on web available which are used by users around the world to share their health related experience with others.

But these data are available in an unorganized way. KnowLife is a large Knowledge Base for health and life science having a wide range of relations about diseases, symptoms, causes, risk factors, drugs, side effects etc. It is a one stop portal which organizes data in a structured and organized way.

In knowLife portal, health related information is inserted with an advanced information extraction method which is so helpful for the physicians and researchers to deepen their knowledge and stay up-to-date with research by searching quickly in an efficient way. This portal will be of great help for our research.

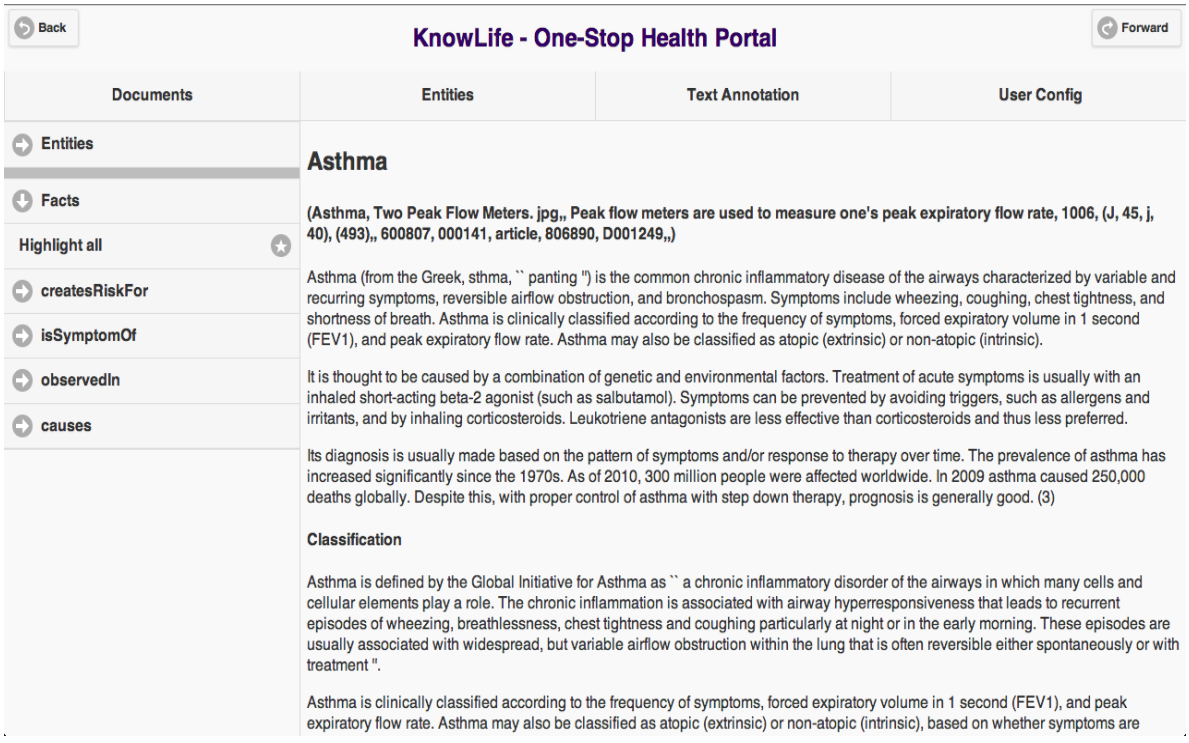

Figure 1: Knowlife Portal User interface

Unlike other Information Extraction methods, KnowLife uses logical consistency reasoning for information extraction method which produces near-human precision. It can extract information from newly published biomedical publications, can also annotate newly seen documents from scientific literature or social media.

Named Entity Recognition and Disambiguation, Pattern Mining, Consistency Reasoning etc. are applied for knowledge harvesting. Other than this, it can also annotate text. Pattern Matching and type checking are used. On pattern matching procedure, Threshold is used to detect a fact candidate. Also, filtering is done on type checking to filter out facts which are not compatible.

As we need to extract information from health communities as well as we need to extract many health and medical related information, there are many web-scale information extraction methods that are available. Traditionally source centric approaches are used for information extraction, here in this research a domain centric approach has been proposed for information extraction which is so efficient and many advantages over source centric approaches. On source centric method, information search and extraction is done only on specific websites. But on domain centric method, it is done on the entire web. This is a huge advantage of domain centric over source centric. This work is helpful for our research too in the aspect of web information extraction.

2.1.3 QMR

It is one of the first tried applications to help in the clinical diagnosis; it provides detailed information and resources that help doctors and clinicians to diagnose the diseases. It provides electronic data bank access to more than 750 common diseases and their complete symptomatology that acts as a decision support tool. QMR knowledgebase includes more than 6,000 clinical signs, symptoms and laboratory findings that describes and explain the disease.

QMR developers claim that all the clinical findings in the QMR database are extensively reviewed by medical experts.

QMR provides functionality to generate extensive DD (differential diagnosis), suggests possible test to diagnose the case, store and manage the case history, QMR developers claim that it is an “expert system” improves medical care by allowing doctors to manage the medical cases more efficiently. The performance of the program is reasonably good, installation and usage is simple; Physicians enter their clinical findings and search for the suggestions and further help, the program processes the physician input comes with the results similar to search engine.

Physician can search by disease for example “Hodgkin lymphoma” is entered then it lists the disease symptomatology, physical signs, lab investigations associated with the disease and differential diagnosis. Developers claim that it is very rare that it returns error however we could not verify and confirm the claim.

QMR also provides list of associated conditions and provides you the details of severity, possible complications and the clinical measures of the disease. However, it was noticed that the systems were missing many possible complications of many diseases.

QMR is developed mainly to provide a medical diagnostic tool; it provides functionality to generate diagnostic hypotheses based on entered clinical signs and symptoms.

The first method is user enters maximum six clinical finding then searches for differential diagnosis to get possible diagnosis.

The second method is user enters complete clinical findings of the patient in response it processes the input and provides notes for each finding. Once a list of differential diagnosis is generated, the physician can apply other program features to the proposed diagnostic hypothesis to refine the diagnosis further. For example, “Finger clubbing” generates list of diagnosis that includes Crohn's disease by double clicking on the disease gives you further details like physical signs, lab tests and its complications.

The program also suggests further input so that physician can get more information from the patient by questioning more and further clinical examinations.

2.1.4 MYCIN

MYCIN was one of earliest diagnosis support systems developed with a short range of functionality operated using simple inference engine with a database of over 600 rules. It is relatively simple diagnostic system and uses simple yes or no questions to get input from the clinician and finally comes up with the possible name of the bacteria. It uses certainty factors as opposed to uncertainty factors and this makes the application fairly simple.

MYCIN usage is simple and limited. Researchers tried the system for therapeutics and they have observed that it suggested relatively correct treatment in about 69% of the cases which was surprisingly better than diagnosing infectious diseases for which the system was originally developed. However, there is no agreed standard for treatment hence the observation was not agreed by many researchers. MYCIN’s strength was in its reasoning approach, it introduced the rule based system development which was used and implemented by many other non-medical domains after MYCIN.

Even though it exceeded the expectations and outperformed the Stanford medical school faculty, it was never actually used in practice for various complex reasons. It covers only small area of internal medicine. Doctors are not convinced that computers can actually diagnose the diseases, and for ethical and legal issues relegated the usage of computers in medical diagnosis. MICIN takes very long time to complete its diagnosis process and this time consumption may be realistic to the physicians. Even though this was technically successful but it has failed to impact on the health care system. The system is not in use anywhere outside the Stanford medical school.

2.1.5 Iliad

Iliad is a diagnostic expert system for Internal Medicine; developing and improving by the University Of Utah School Of Medicine's department of Medical Informatics for last two decades. The system supports more than 5000 clinical findings and provides reasonably accurate diagnosis for more than 1,500 medical conditions.

One of the important features that Iliad offers is the ability to analyze a particular patient's case and to determine the most cost-effective method for diagnosing and treating the patient. Iliad was developed originally for the Apple Mac; and a version for the PC running windows has also been released. Iliad is primarily used as a teaching tool for medical students. This helps the students to improve their skill in differential diagnosis. A clinical case can be simulated through this system and students have to diagnose the case. Students can query Iliad for useful patient history, physical examinations, or required laboratory investigations for the patient. Iliad process the query and evaluates alternative decision strategies with the use of "best Information Algorithm" this is combination of content, weightage and the cost.

Process result then provides alternative work-ups in the order of cost-effectiveness Iliad is developed based on Bavesean logic and Boolean knowledge frames to illustrate disease in internal medicine. The frames allow the use of sensitivities, specifics, and rules to describe the relationship between disease and its symptomatology and provides a basis for Iliad logic.

2.1.6 Internist-I

Internist-I is a broad based clinical diagnosis support systems and the major contributors for the development of the project include Randolph A. Miller, Harry E. Pople, and Victor Yu. It was originally developed for cases in Existing Clinical Diagnosis Software Systems 37 internal medicine.

Internist-I was core part of "The Logic of Problem-Solving in Clinical Diagnosis" course in university of Pittsburgh for nearly 10 years. With the help of medical experts the fourth year medicos in university of Pittsburgh has been entering and updating the medical data in to the system.

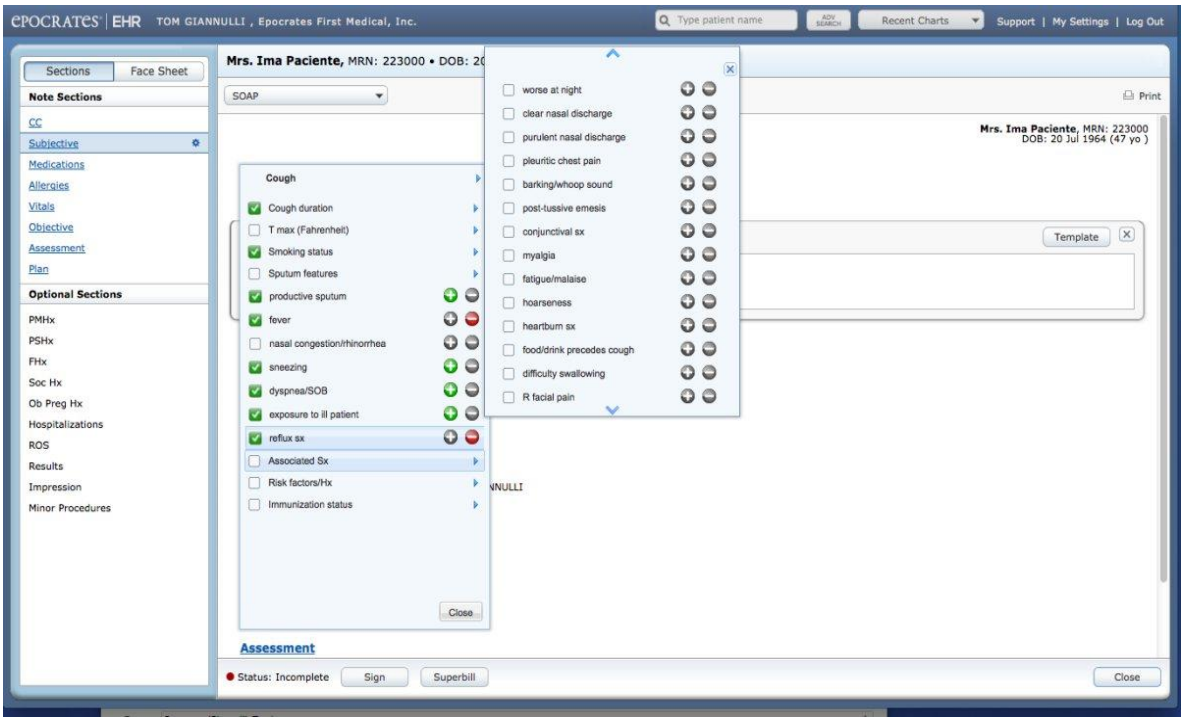

Figure 2: Internest-I user interface

They encoded the clinical and pathological finding and standard medical reports in to the system. By 1982 INTERNIST-I project had fifteen person years medical data entry, and covered 70 to 80% all the possible diagnosis in the medicine. Information stored in the system includes symptoms and signs, laboratory investigation results, and the patient's case history.

Internist-I did not follow the traditions of other systems instead it used the powerful ranking system. It ranks clinical findings in relation to the disease and it ranks disease itself depending on its occurrence. It also uses heuristic rule based partition algorithm to create problem area and exclusion functions to eliminate diagnostic possibilities.

These rules create list of diagnosis in probable ranking order. When input data is not enough to suggest the diagnosis then system asks for further information or further examinations to resolve the case. Some documentation claims that Internist-I works better if the clinical finding of the patient is related to the one disease but other documentation disputes the claim and claims that it handles very complex cases very well.

2.1.7 DXplain

DXplain has been in use for the last two decades. It has evolved and gained some popularity over the time. First version was developed in 1984 with illustrations of about 500 common diseases and it was released in 1986. Further versions were released in 1987, 1090, 1991, 1095 and 1996 with deceases and functionality. Since 1996 DXplain has been completely web

based. DXplain is a clinical decision support system and it functions in two modes, electronic medicine book and a medical reference system or case analysis mode.

In reference or case analysis mode, it accepts patient's clinical data like signs, symptoms, and laboratory findings and processes the data and produces the list of probable diagnosis in an order. It also provides logical reasoning for each of the diagnosis and why it was considered so that the physician/student can explore more regarding its manifestations. In medical textbook mode, DXPlain provides illustrations of over 2300 diseases and it explains the signs and symptoms of each disease. It also provides epidemiology, etiology (cause of the decease), pathology, complications, and the prognosis of the disease. In addition, it also provides up to ten references for each disease and these references provide more information, reviews and research information regarding the disease.

The current version of DXplain includes over 2300 diseases and over 4900 clinical manifestations (symptomatology, physical signs, epidemiology, laboratory investigations and other modern investigation findings like endoscopy, CT-Scan and MRI findings). Every disease consists minimum 10 clinical findings to maximum 100 clinical findings. Each clinical finding is related to one or more diseases and with the frequency of its appearance in the disease.

There are over 230,000 data relationships between a clinical finding and a disease. Each clinical finding has 1 to 5 disease independent rating to indicate its significance. Each disease also has two related values crude approximation and prevalence and disease also ranked between 1 and 5 based other reasons.

2.1.8 Isabel

Isabel is a widely used web based clinical diagnosis support system, Isabel accepts either key clinical findings or whole text entry of the clinical case and processes the request by using novel search strategy and identifies probable diagnosis from the given clinical findings. The physician can enter unlimited clinical conditions or complete case to find the probable diagnosis. The program also includes the data dictionary of the medical terms and clinical conditions and the library includes six medical textbooks and 49 major medical journals. The search results are filtered on epidemiological findings geographic location, age, sex and hobbies and system then displays more than 30 probable diagnoses. Up to ten diagnoses are presented on first webpage with web links. Physician can then explore each disease by clicking the link, to see other possible diagnosis; physician can click more diagnosis link.

Isabel uses natural language processing and search algorithm that searches clinical data in the database system and comes up with new 30 diagnoses. However exact algorithm that Isabel uses is undisclosed and company does not want to reveal it.

Figure 3: Isabel Healthcare User Interface

Recent release of Isabel is called Isabel PRO and it has two major components Isabel PRO Diagnosis Reminder System (IDRS) and Isabel PRO Knowledge Mobilizing System (IKMS).

Isabel Pro is currently available for hospitals and poly clinics interfaced with high profile electronic medical records (EMR) systems in the USA, Isabel provides input filed for age, sex, and clinical conditions and a “Suggest Diagnosis” link button to the process the data. It returns results in separate window when the “Suggest Diagnosis” link is clicked. User can explore and refine the diagnosis by entering more finding and clicking relevant links. It also provides additional intelligent layer suggest more options to the clinician. Diagnosis is a life critical process and searching text for isn’t appropriate for life critical systems. Even though it gives somewhat better results comparing with other systems, it may not make very big impact on clinical diagnosis and health care. Most of the existing system can diagnose the cases up to certain extent but they are not mature enough to use in life critical environment, they need to be evolved further with above 99% accuracy with better algorithms and with better data store.

2.1.9 Baidus “Medical Robot”

A majority of Chinese online turn to the Web first for health information, and voice search is far

Wei's (a researcher for Baidu in Sunnyvale, Calif.) project, called AskADoctor in English, is one of the earliest to emerge from Baidu's deep-learning division since it hired Andrew Ng, a renowned data scientist and former marquee researcher at Google. And it's an example of the unique tech interface the company can produce given its privileged access to the world's biggest nation, which has kept Silicon Valley giants at arm's length.

The initiative is also another sign of the broader industry trend of tech firms storming into medical sciences with their artificial intelligence guns drawn. Earlier this week, IBM announced plans to acquire medical imaging company Merge Health, turning its data over to IBM's supercomputers. Google, while not fully public about its medical programs, has similar ambitions. Apple has its wearable health strategy.

Baidu's advantage comes with scale. Ng's team talks of delivering research that has a direct impact on the company's bottom line, reaching "hundreds of millions" of users. China's tremendous Internet population makes the latter goal easier.

The team is also betting big on voice, a field where it may advance more in China than other rivals. Since February, Baidu's deep-learning stateside team, around 40 researchers, has worked on building artificial neural networks to process Mandarin. The technique allows machines to render the language — a complicated one, as it's tonal and character-rich — with far more computing power. (See here for an explanation of neural nets and how tech giants are deploying them.)

With AskADoctor, the computer voice translation couples with another deep-learning model that ropes in the health data owned and scraped by Baidu across the Chinese Web. Wei said the product can assess 520 different diseases, representing upward of 90 percent of the most common medical problems nationwide. A desktop version is now available, and Baidu plans to release the mobile app soon. Over time, Wei added, Baidu hopes to tie the product in with medical records in China, which are currently in the early stages of going digital.

The product fits with the company's new focus on connecting online users to offline services — eventually, it will take a cut when it connects users to local doctors. It's a necessary pivot, an attempt to reinsert the search engine's relevancy as app usage outpaces the mobile Web. But it's a costly one: Last quarter, Baidu reported revenue of $2.7 billion, below expectations, and said it plans to invest $3.2 billion in online-to-offline services.

Ng's AI has helped counteract those costs, according to Baidu. A computer vision-driven improvement to an image product for advertisers improved click-through and paid-click rates, the company said on its earnings call.

The AI team has also brought a headache. In June, Baidu was barred from an international AI competition, in which companies like Google, Facebook and Microsoft compete, for breaking the rules with its image-recognition tech. Ng led the prompt move to fire Ren Wu, the researcher Baidu faulted for the breach, but the incident has damaged the company's standing in the insular research world.

Baidu did not comment much on the episodes; beyond that it had let go of the staff responsible.

Asked what sets Baidu's AI division apart, Ng returned to size, and not just China's. Baidu is investing heavily in AI hardware — it clusters large numbers of graphics processing units trained on speech models — something Ng may not have had at Google, which tends to favor a more dispensable approach to hardware.

2.2 Related Works: Strength and Weakness

The work presented in [1] focuses on disease prediction from clinical data provided by New York - Presbyterian Hospital. As these are clinical data, automated disease prediction is relatively different and easier than predicting from user text input. It is observed that input from common user contains less number of clinical terms. That means, matching the symptom names from user input with system database has more complexity.

[3] emphasizes on prediction of potential infectious disease outbreaks from online text sources. Which is also a specialized source where explicit medical terms are used.

A lot of effort is put on to predict specific diseases [6], [15]. E.g. [6] focuses on predicting coronary heart diseases by mining text. There are also quite a number of research works that have been done in recent years on healthcare forums. [7] is such a work where natural language processing is used to rate and analyze user comments in order to predict diseases and extract rare side effects of drugs. This system took into account suggestions provided by different users on comment sections in disease analysis.

Healthcare websites such as isabelhealthcare.com, mayoclinic.org, patient.co.uk, are providing disease prediction based on user input ([13], [14] and [16]). [14] uses jargon-laden interface (I.e. users need to navigate through a long-list of symptoms). From user's point of view, it is a cumbersome task and the process is time consuming as well. Moreover, if a certain symptom is not found by the users, they are compelled to skip that symptom which is not desired at all. [13], [16] take guided input from user. However, they rely on mere symptom-disease relationship framework ([17], [25]). Upon user input, these systems start looking for exact word match in the database from each input line. Thus it does not allow linguistic diversity. E.g. if the database does not contain a symptom's synonym used by a user, it will not be able to match the input perfectly. If the input contains more non-technical terms than expected, its performance degrades significantly. The framework used is very much rigid and confined to specific input types.

Dx-Plain, Internist-I, Iliad, MYCIN and QMR are rarely used today. The widely used one is Isabel Healthcare system. But Isabel Healthcare uses text database. And so during their searching procedure, there are many false matches. Also its performance degrades greatly with more non-technical terms. Though Isabel claimed that it has 95% accuracy, but 95% accuracy is not enough in diagnosis decision support system. It should be upto 99%. On some researches, it is found that accuracy of Isabel is much lower than 95% and it has many false detection of diseases.

Chapter 3

PROPOSED ARCHITECTURE

3.1 MOTIVATION

We propose a novel architecture (Automated Disease Prediction System (ADPS)) to predict diseases automatically based on user input on symptom checkers.

We presume that the user will give text input in one sentence describing a single symptom at a time (guideline for user input). Subsequent symptoms can be added in new lines. After getting user input, the system will scan through each line and tag each word according to their relevant parameter. Then after performing certain computations (to be described later) the system will return a list of possible diseases ordered according to the likelihood of their occurrences.

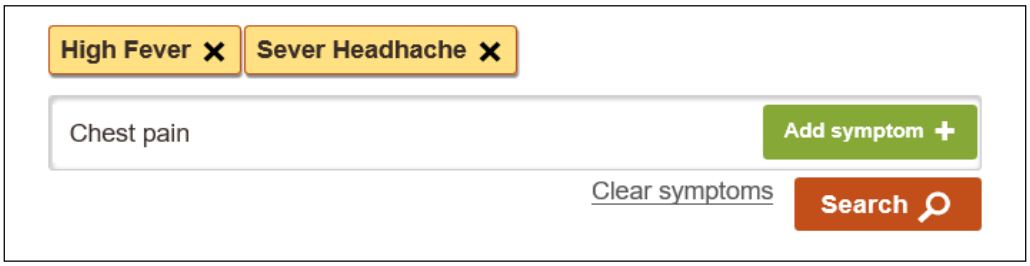

Figure 4: Overview of user input

3.2 RELATIONAL DATABASE

Instead of the text database, we have modeled a relational database which can be considered as a tree based disease symptom-database. It can be viewed as a bipartite graph.

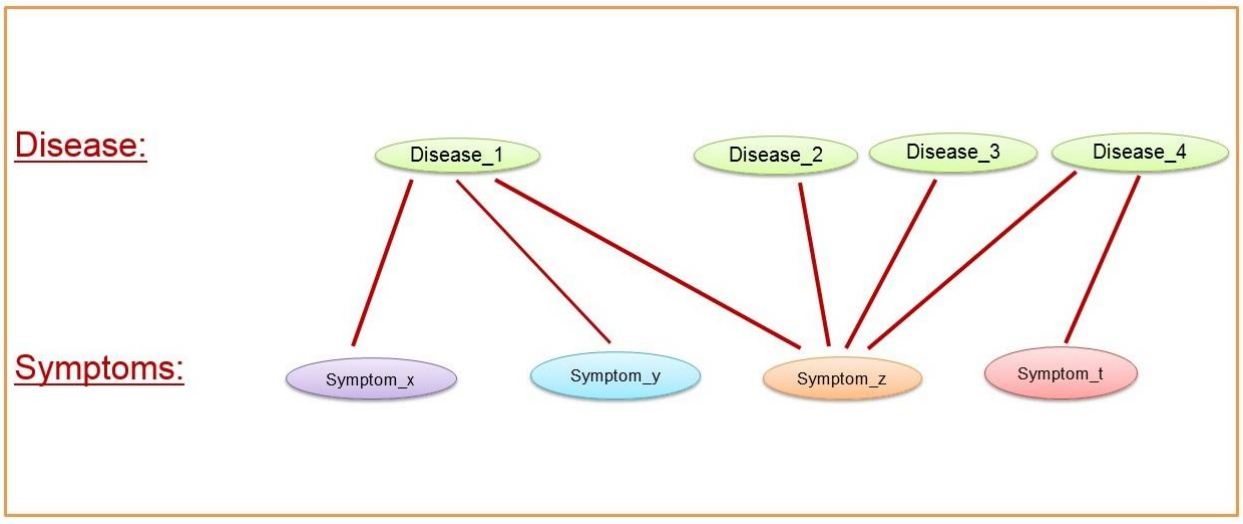

Figure 5: Disease-Symptom tree

Here disease and symptoms are nodes. They can be connected with edges. An edge between a disease (D) and a symptom (S) means S is a symptom of disease D.
The graph is bipartite. So there can't be any edge between one disease with another nor between one symptom and another.

3.3 ADPS COMPONENTS

The components of ADPS architecture is described on the following sections.

3.3.1 Relevant Attribute (RA) Data Structure

Most of the existing disease prediction systems ([3], [13], [16]) where user input is taken as text, focus only on symptom-to-disease relationships. Associating a disease merely based on a symptom name can significantly decrease the accuracy of disease prediction. Because there are other parameters that can help pin pointing a disease more accurately. E.g. High fever is a symptom of dengue while mild fever is a symptom of Reiter's syndrome or reactive arthritis. Here if the intensity is not taken into consideration then only 'fever' can refer to either one of these two diseases. Similarly, time can also be a vital parameter to be considered in case of disease prediction. For instance, high temperature at 'night' is a symptom of respiratory tract infection (cold). Here timing (night) of the fever cannot be ignored. If neglected, the accuracy of disease prediction can deviate significantly, ultimately leading to incorrect prediction.

In this work we propose RA data structure where five relevant parameters from user input are taken into account and these parameters will be proven vital in accurate disease prediction in subsequent sections. RA data structure is as follows.

General Form: < S, T, I, O, D >

S = Symptom name (Fever, Headache etc.)
T = Time (Morning, Night etc.)
I = Intensity (Severe, Mild etc.)
O = Organ name (Abdomen, Head, Heart etc.)
D = Duration (10 days, 1 month etc.)

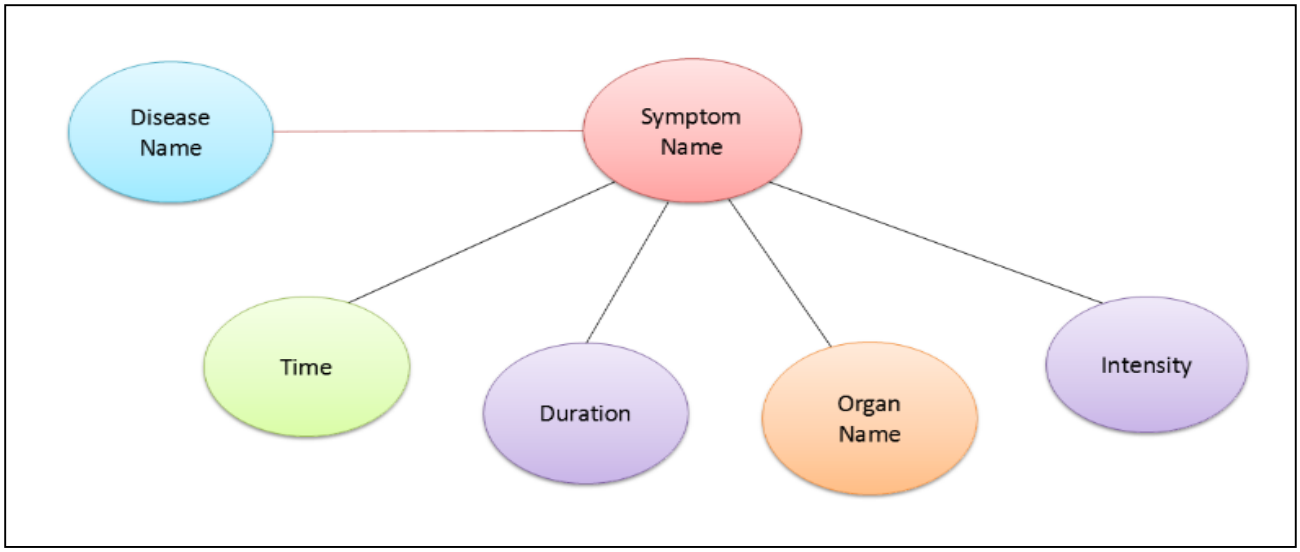

Figure 6: RA data structure

3.3.2 Disease Symptom Database

It is a disease symptom database developed from expert sources ([11], wikipedia) where each disease is associated with 5 parameters (S, T, I, O, D) of RA data structure. E.g. figure 3 and 4 is a logical overview of the database.

S	T	I	O	D
Fever	×	High	×	×
Headache	×	Severe	×	×
Pain	×	×	Eyes	×
Pain	×	Severe	Joint	×
Pain	×	×	Muscle	×
Fatigue	×	×	×	×
Nausea	×	×	×	×
Vomiting	×	×	×	×
Rash	×	×	×	×

Figure 7: DB representation For Dengue (Matrix D-D)

S	T	I	O	D
Fever	×	High	×	×
Headache	×	×	×	×
Pain	×	Severe	Abdomen	×
Pain	×	×	Muscle	×
Fatigue	×	×	×	×
Dry Cough	×	×	×	×
Vomiting	×	×	×	×
Rash	×	×	×	×
Diarrhea	×	×	×	×

Figure 8: DB representation for Typhoid (Matrix D-T)

3.3.3 WORD TAGGING

Initially each word is tagged according to RA data structure. From each input line, words will be tagged according to their correspondence with symptom name, time, intensity, organ name and duration.

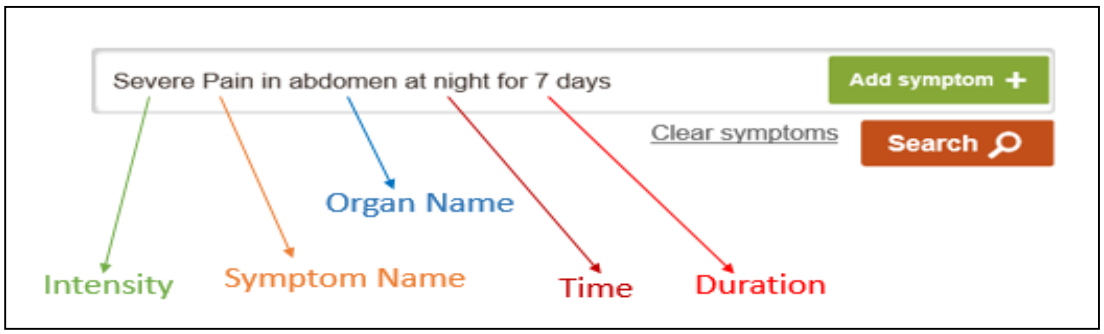

Figure 9: Word Tagging

Tagging will be done using following three techniques:

i) Synonym Parent Tree
ii) Symptom Reference Tag & Decision Tree
iii) Relevant Attribute (RA) Array

3.3.4 Synonym Parent Tree

User input can have great linguistic diversity. Same thing can be described using different words. Also people can use synonym of a word. Therefore, it is very likely that the user's input will often not be an exact match to what we have in our database.

Words like ‘urinating’, ‘urinate’ and ‘urinated’ represent something related to ‘urination’. When input words are matched with database, many words may be returned as unmatched words in spite of having the same meaning. To tackle such cases, we propose the use of a Symptom Parent Tree. Here each word is pointed to its root or parent word. Each child is a synonym of its parent. If any of the trees contain a matching child word, the input word is replaced with the root of the matched tree.

Each word is parsed from the input and this is how whenever it is possible a word is rectified so that it resembles the exact same database entry.

After this word modification step, each word is searched against the database entries to find the corresponding parameter name. E.g. consider a word ‘severe’ in a user input line. The database has three types of intensity values: high, medium and low. ‘Severe’ corresponds to ‘high’, therefore synonym tree converts the word ‘severe’ to ‘high’.

Then the word ‘high’ is looked up in the database and it is found that the parameter name of ‘high’ is Intensity. So the word ‘high’ gets the tag Intensity according to RA data structure.

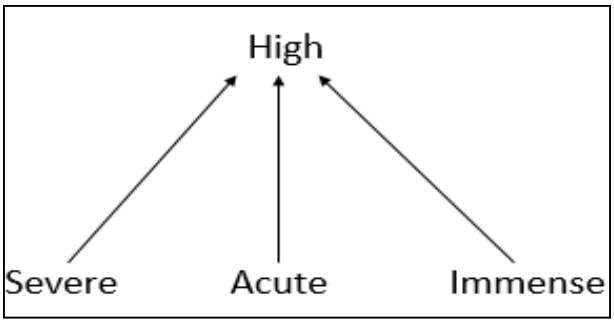

Figure 10: Synonym Parent Tree

3.3.5 Symptom Reference Tag & Decision Tree

For accurate disease prediction, each word is required to be tagged correctly. Symptom parent tree approach is not enough to fulfill this goal.

A single symptom name may often be comprised of more than one word rather than single clinical word. As the user can express the same thing in different ways, identifying a specific symptom can be very tricky at times. For example, a user might not use the word ‘insomnia’ to describe the fact that he is experiencing difficulty with sleeping or having insufficient sleep. Instead he may write “I cannot sleep at night”. However, our approach should still be able to interpret it as ‘insomnia’ even though the exact user input is not part of the database.

To cater for the above mentioned scenario, we propose a decision tree based solution to determine the symptom name from such compound inputs. To use the decision tree, a symptom associated tag is introduced. We call this ‘symptom reference tag’. For all possible symptoms, there are related tags associated with it in the database. For example, the related tag for ‘Insomnia’ is 'sleep'. This implies, if the user does not specifically use the word ‘insomnia’, he is expected to use the word sleep somewhere in his input to refer to the fact that he is having trouble sleeping. Using decision tree, symptoms from a text input can be found. Traversing the decision tree along either Sleep --> Deficiency (If input line contains negation) will ultimately lead us to ‘Insomnia’ as being the symptom. Likewise, if the decision tree is traversed along Sleep --> Excess, ‘Hypersomnia’ will be detected as the relevant symptom.

Symptom Name	Reference Tag
Insomnia	Sleep
Hypersomnia	Sleep

Table 1: Reference tag example

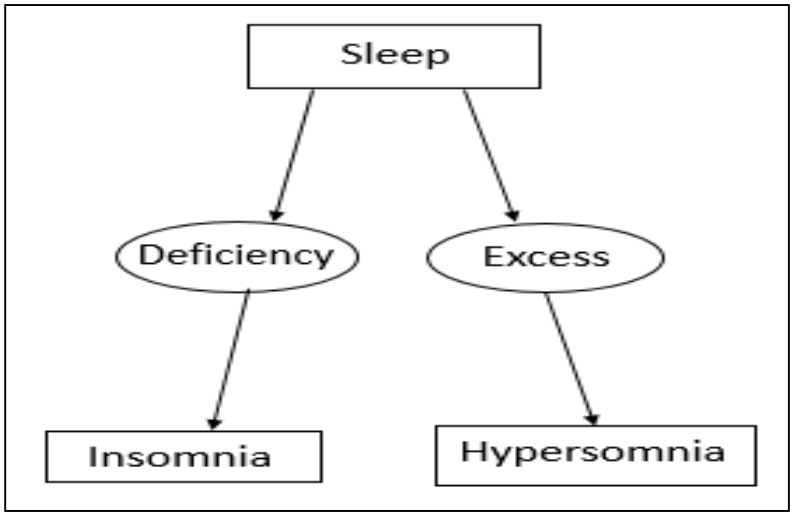

Figure 11: Decision Tree

3.3.6 Relevant Attribute (RA) Array

Once the type of each of the input words is determined using techniques described in section V and VI, words will be put on to 5 different arrays which we named as RA arrays.

If an input word is a symptom name, it will enter the symptom array. Likewise, if an input word represents the intensity of a symptom, it will enter the intensity array and so on.

```
function word_tagging ( string input)
    for each word
          change a word to its synonym parent word (if any)
         check the word in database
         if the word is found in database
              then put it in relevant parameter array
        else If not found
              search in symptom reference tag table
              if  a reference word found,
                   then traverse relevant decision tree
                   if result is found
                      then put it in relative RA Array and continue
                  end if
             end if
        end if
   end for
```

Figure 12: Algorithm for word_tagging

As far as the algorithm and RA data structure are concerned, any input word whose type cannot be determined is deemed to have no apparent significance and thus will be discarded.

The contents stored at the same index of different arrays will have relevance i.e. if those five arrays are

- symptom[]
- time[]
- intensity[]
- organ[]
- duration[]

Then if intensity[n] denotes 'High' intensity, it will refer to the symptom of the n^{th} index of the symptom array i.e. symptom[n]. For example, if symptom [n] = 'Fever' and intensity [n] = 'High', then 'High' denotes the intensity of the symptom 'Fever'.

Arrays will grow in size with each separate symptom input from user. E.g. if the user enters 4 symptoms, each of the arrays will have 4 elements. It may be noted that all of the arrays except the symptom name array can hold null (x) values where a null entry indicates the absence of a relevant detail, since it is understandable that each and every symptom may not have all five parameters (E.g. 'high fever' does not associate any organ name).

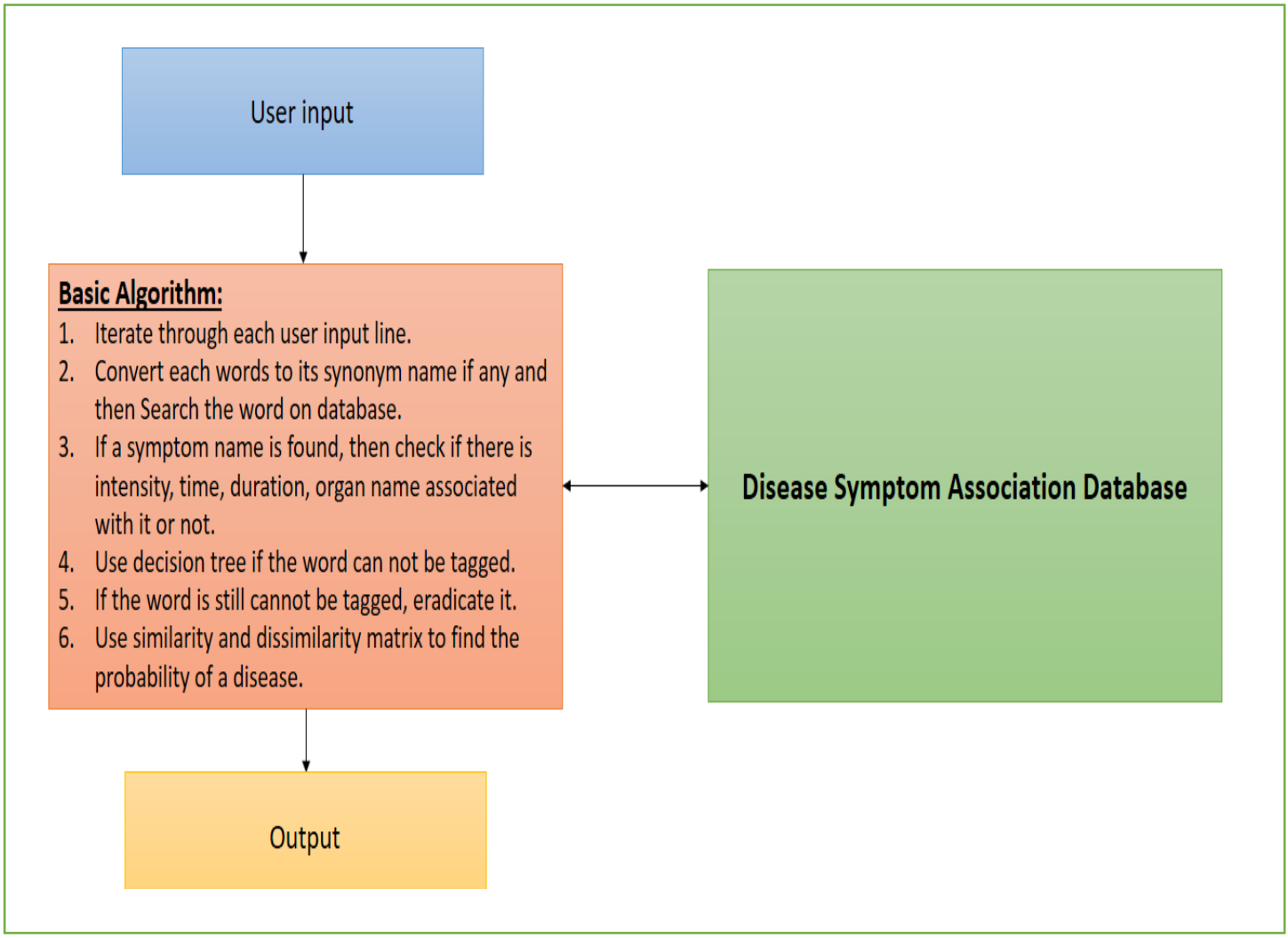

Figure 13: Workflow

CHAPTER 4
Evaluation and Results

4.1 PROBABILITY COMPUTATION

'Walk along an example' approach will be convenient to understand the computation process of disease prediction in ADPS.

Here is a set of user query:

1. I have severe fever.
2. Suffering from headache.
3. Muscle pain.
4. Vomiting.
5. Pain in joints.
6. Rash.
7. Fatigue.

According to RA data structure, for this example query, 5 arrays are required where each of the arrays will have 7 elements (0 - 6) to store the tagged words. After scanning through the 7 input lines the contents of the arrays will be as follows:

S[0] = 'fever'	T[0] = '×'	I[0] = 'high'	O[0] = '×'	D[0] = '×'
S[1] = 'headache'	T[1] = '× '	I[1] = '×'	O[1] = '×'	D[1] = '×'
S[2] = 'pain'	T[2] = '×'	I[2] = '×'	O[2] = 'muscle'	D[2] = '×'
S[3] = 'vomiting'	T[3] = '×'	I[3] = '×'	O[3] = '×'	D[3] = '×'
S[4] = 'pain'	T[4] = '×'	I[4] = '×'	O[4] = 'joint'	D[4] = '×'
S[5] = 'rash'	T[5] = '×'	I[5] = '×'	O[5] = '×'	D[5] = '×'
S[6] = 'fatigue'	T[6] = '×'	I[6] = '×'	O[6] = '×'	D[6] = '×'

From the above mentioned arrays a Data Matrix will be generated like the following one.

S	T	I	O	D
Fever	×	High	×	×
Headaches	×	×	×	×
Pain	×	×	Joint	×
Pain	×	×	Muscle	×
Vomiting	×	×	×	×
Rash	×	×	×	×

Figure 14: Matrix Dq

Initially symptoms from this data matrix are fetched and mapped with the symptoms in the database. Then data matrices corresponding to all diseases that have at least n (n is defined as 3 in this work) symptoms matched against the query data matrix are recorded for further processing.

In this case, matrices in figure 3 and 4 are fetched/retrieved from database named D-D and D-T (See section V(B)).

In the next step 'asymmetric binary similarity' [23] factor is calculated among the user query data matrix and matched data matrix/matrices by the following equation.

$$Sim(mat\_i, mat\_j) = q / (q+r+s) \quad \text{---------- (I)}$$

Where,

q is the number of attributes that equal 1 for both objects,

r is the number of attributes that equal 1 for object i but equal 0 for object j,

s is the number of attributes that equal 0 for object i but equal 1 for object j.

As database fetched matrices are verified as true (to be described later), values present in these matrices are considered as 1, and others are 0. If matrix size is not same for user query data matrix (Dq) and DB fetched data matrix, we consider the empty rows as complete mismatch.

Here,

sim (Dq, D-D) = q / (q + r + s) = 26/36 = 72.22 %

sim (Dq, D-T) = q / (q + r + s) = 23/36 = 63.89 %

It is clearly observable that probability of occurring Dengue is higher according to user input.

4.2 EVALUATION AND ACCURACY

As stated before, ADPS provides disease predictions in ascending order like other existing systems. We classify the ranking in 3 clusters.

If the probability is between 1 to 50% (inclusive), we consider it as low probability (L).

If the probability is between 51 to 100% (inclusive), we consider it as high probability (H).

We use Visual studio 2015, Oracle Database as our simulation software. Language was C#.

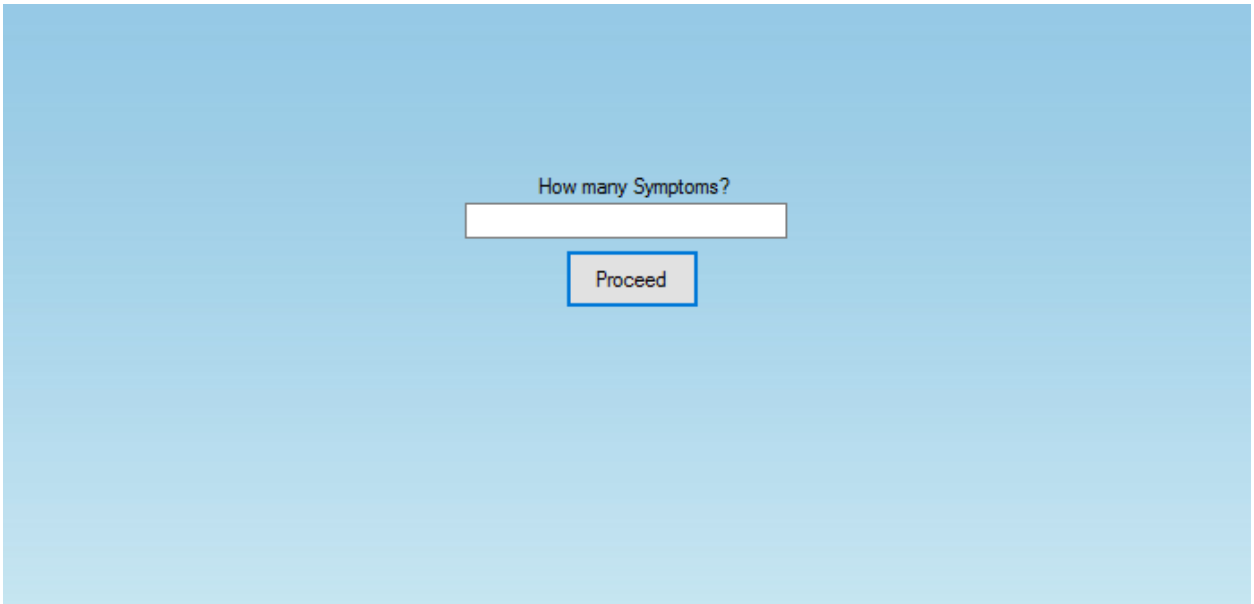

Figure 15: Evaluation software (a)

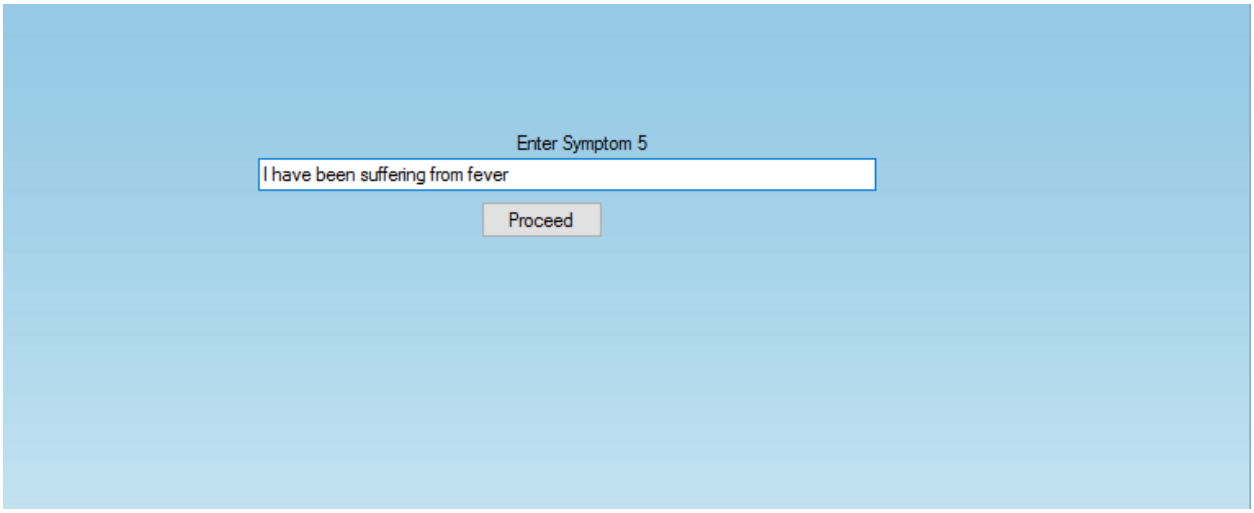

Figure 16: Evaluation software (b)

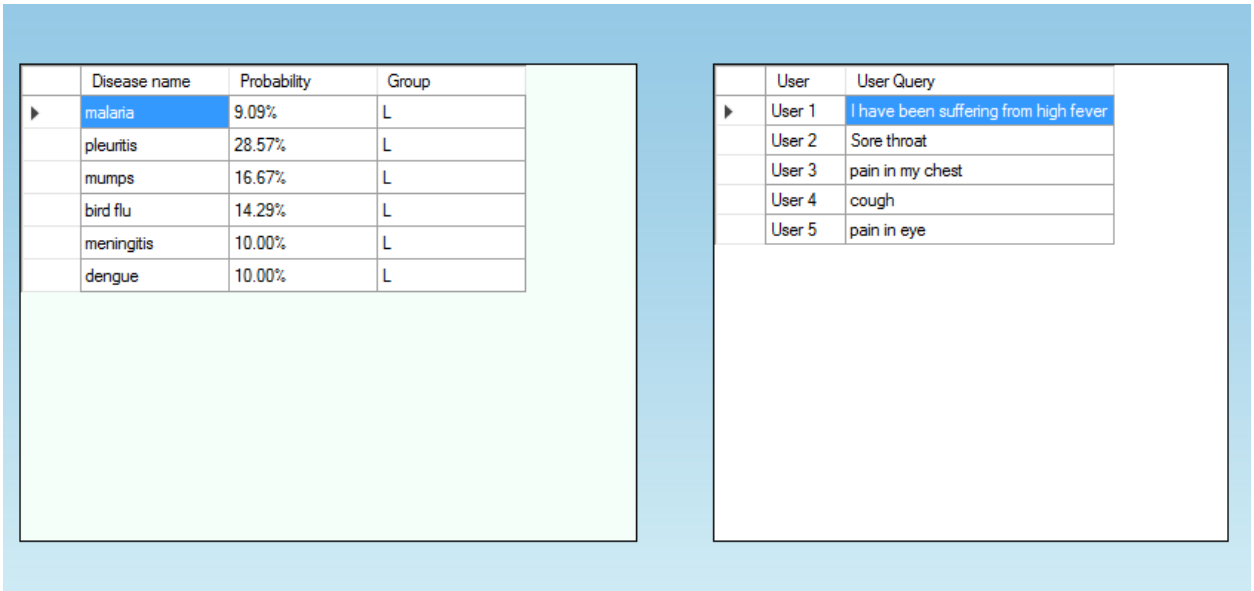

Figure 17: Evaluation software (c)

To compare their relative accuracy, we check each of the ranked predictions against the ground truth. The ground truth symptom-disease associations are recorded from Wikipedia. To better understand the accuracy comparison process let us consider an imaginary data set (symptom list input from a patient) where, 5 diseases fall in high probability cluster, 3 in medium and 1 in low. Considering it as ground truth, let us take a look at the following table:

Disease	Ground Truth	ADPS prediction	Normal Disease Symptom prediction [17]
D1	H	H	H
D2	H	H	L
D3	L	H	H
D4	H	H	H
D5	H	H	L
D6	L	L	L
D7	L	L	L
D8	H	L	L
D9	H	H	L

Table 2: Ground Truth Comparison Table

To compute accuracy values of each column (ADPS & normal) are checked against the ground truth. Intuition says that each checking will produce binary values (0 for mismatch & 1 for match). If the difference between two clusters is of degree 2, then it is considered as complete mismatch (0). However, if the difference between two probability cluster is just of one order (L-> M -> H i.e. 0 -> .5 -> 1 e.g. if ground truth is given 'H' and the result shows it as 'M') then it should not be considered as just a full mismatch (0) rather a half match (.5).

Accuracy = m/t ---------- (II)

m= cumulative match factor

t= total number of diseases

ADPS Accuracy = 8/9 = 88.89%

Normal Accuracy = 5/9 = 55.56%

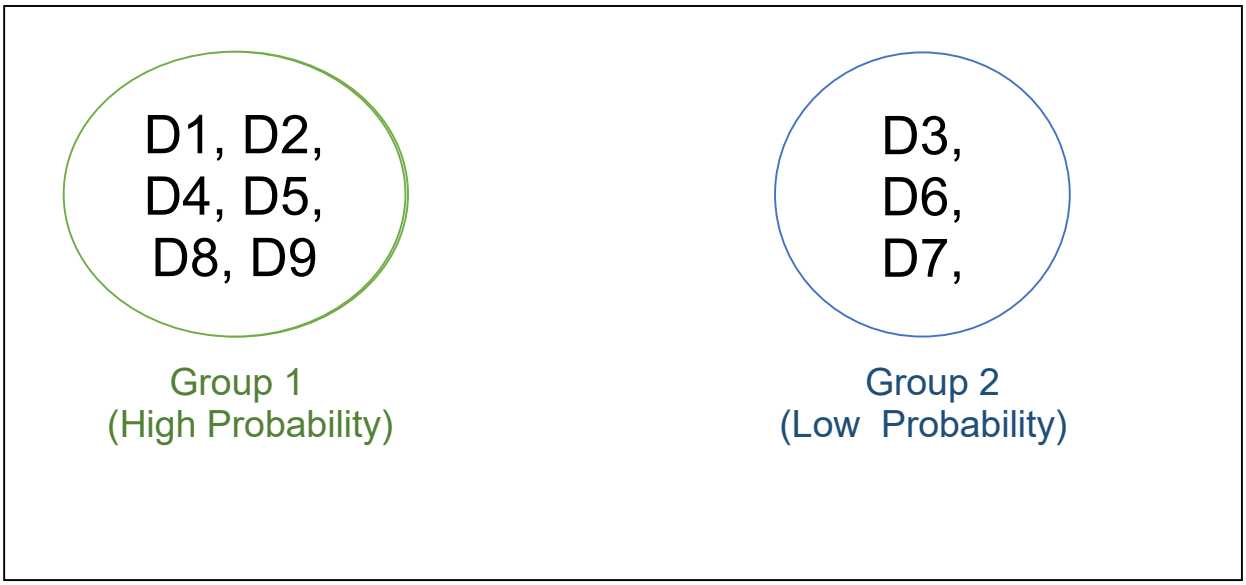

Figure 18: Two groups

This accuracy value resembles the quality of a predicted disease ranking list by a system (higher value means more accurate). It is vital because the occurrence probability of some lower ranked diseases cannot be ruled out as many diseases share a number of common symptoms.

In order to test the effectiveness of our approach, we have picked 10 user queries [24].

The results produced by ADPS (Using equation I) and disease-symptom matching system are then arranged in tabular form like table 2 for each disease.

The accuracy for each disease is determined (using equation II) and results are shown in the following table:

Experiment	Accuracy in disease symptom matching system [17]	Accuracy using ADPS	Improvement
E1	69.82%	81.61%	16.88%
E2	78.95%	91.42%	15.79%
E3	71.4%	81.2%	13.67%
E4	51.2%	73.3%	43.07%
E5	64.67%	73.38%	13.47%
E6	69.56%	85.7%	23.27%
E7	58.72%	71.4%	4.60%
E8	76.13%	91.75%	20.52%
E9	65.2%	81.5%	25.97%
E10	65.7%	83.3%	26.88%

Table 3: Accuracy from 10 user input

An average of 20.41% higher accuracy is observed after evaluation with a minimum of 4.60% and maximum of 43.07%. It is worth noting that ADPS accuracy is significantly better. Therefore, disease prediction is more accurate in ADPS.

CHAPTER 5
CONCLUSION

5.1 FUTURE WORK

We will try to improve our database with more data. We will try to make our system more efficient to non-medical terms. We will allow voice commands along with input text to predict diseases.

5.2 CONCLUSION

Technology has ushered numerous ways to drive mankind towards a better world, a better life. People will be better off if technology is blended into our lifestyle. In this work, we show that our 'Automated Disease Prediction System Architecture' can help people who are facing difficulties, better understand their physical condition by predicting potential diseases. We also show that our framework enables the system perform significantly better than existing ones. Having said that, our system accuracy can be increased further as there is space left for improvement. Like the decision tree and parent tree generation is a cumbersome task but it is a continuous process, same goes with the enrichment of the database. It will get better and better over time and accuracy of disease prediction will also be on the rise.

REFERENCES

[1] Xiaoyan Wang, Amy Chused, Nomie Elhadad, Carol Friedman, and Marianthi Markatou : "Automated Knowledge Acquisition from Clinical Narrative Reports." , AMIA 2008 Symposium Proceedings, pp : 783-787.

[2] Pew Research center health fact sheet : www.pewinternet.org/fact-sheets/health-fact-sheet.

[3] Nicolae Dragu, Fouad Elkhoury, Takunari Ralph and A. Morelli Nicolas di Tada : "Ontology-Based Text Mining for Predicting Disease Outbreaks." , Proceedings of the Twenty-Third International Florida Artificial Intelligence Research Society Conference (FLAIRS 2010)

[4] http://edition.cnn.com/2012/05/04/tech/social-media/facebook-lies-privacy

[5] R. Tamilarasi and Dr. R. Porkodi: "A Study and Analysis of Disease Prediction Techniques in Data Mining for Healthcare." , International Journal of Emerging Research in Management and Technology march, 2015. ISSN: 2278-9359 (Volume-4, Issue-3).

[6] Kumar Sen, Shamsher Bahadur Patel and Dr. D. P. Shukla : "A Data Mining Technique for Prediction of Coronary Heart Disease Using Neuro-Fuzzy." , International Journal Of Engineering And Computer Science ISSN 2319-7242 Volume 2 Issue 9 Sept, 2013 , pp : 2663-2671.

[7] Subhabrata Mukherjeey, Gerhard Weikumy and Cristian Danescu : People on Drugs: "Credibility of User Statements in Health Communities." , Proc. of the 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2014.

[8] Slav Petrov, Dipanjan Das and Ryan McDonald: "A Universal Part of-Speech Tagset." , Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC 2012).

[9] Subhabrata Mukherjee, Pushpak Bhattacharyya : " Sentiment Analysis in Twitter with Lightweight Discourse Analysis." , 24th International Conference on Computational Linguistics 2012.

[10] Samaneh mogaddem : "Beyond Sentiment Analysis : Mining Defects and Improvements from Customer Feedback.", 37th European Conference on IR Research, ECIR 2015, Vienna, Austria, March 29 - April 2, 2015. Proceedings.

[11] Patrick Ernst, Cynthia Meng, Amy Siu, Gerhard Weikum : “KnowLife: a Knowledge Graph for Health and Life Sciences.” 30th International Conference on Data Engineering (ICDE), 2014 IEEE, pp : 1254 - 1257.

[12] F. Suchanek,G. Weikum : “Knowledge harvesting from text and Web sources.” Conference: IEEE 29thInternational Conference onData Engineering (ICDE), 2013.

[13] www.isabelhealthcare.com [Accessed 12/10/2015]

[14] www.mayoclinic.org [Accessed 17/10/2015]

[15] Saba Bashir, Usman Qamar, Farhan Hassan Khan: “ BagMOOV: A novel ensemble for heart disease prediction bootstrap aggregation with multi-objective optimized voting Received.”

[16] www.patient.co.uk [Accessed 11/10/2015]

[17] www.symptomchecker.isabelhealthcare.com [Accessed 30/10/2015]

[18] Suchanek and Gerhard Weikum: “Knowledge Harvesting from Text and Web Sources. Fabian.” , ISBN: pp : 1250-1253

[19] www.webmd.com [Accessed 22/10/2015]

[20] Samaneh Moghaddam, Martin Ester: “The FLDA model for aspect-based opinion mining: addressing the cold start problem.” , 22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013.

[21] Samaneh Moghaddam, Martin Ester: “On the design of LDA models for aspect-based opinion mining” , 21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012.

[22] Samaneh Moghaddam, Martin Ester: “Aspect-based opinion mining from product reviews.” , The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012.

[23] Data Mining Concepts and Techniques, Third Edition: Jiawei Han, University of Illinois at Urbana–Champaign and Micheline Kamber Jian Pei, Simon Fraser University.

[24] Mount Adora Hospital & Diagnostic Center, Mirboxtula, Nayashark, Sylhet-3100.

[25] www.bettermedicine.com [Accessed 15/10/2015]

[26] http://www.kiranreddys.com/articles/clinicaldiagnosissupportsystems.pdf [Accessed 30/10/2015]

[27] http://patient.info/forums [Accessed 27/10/2015]

[28] Amit X. Garg, MD; Neill K. J. Adhikari, MD; Heather McDonald, MSc; M. Patricia Rosas-Arellano, MD, PhD; P. J. Devereaux, MD; Joseph Beyene, PhD; Justina Sam, BHSc; R. Brian Haynes, MD, PhD: "Effects of Computerized Clinical Decision Support Systems on Practitioner Performance and Patient Outcomes" , JAMA. 2005;293(10):1223-1238. doi:10.1001/jama.293.10.1223.

Acknowledgements

- Dr. Abu Raihan Mostofa Kamal, Associate Professor.
- Nafiul Rashid, Lecturer.