**A semester project**

**report on**

**Summarization of Medical Blogs**

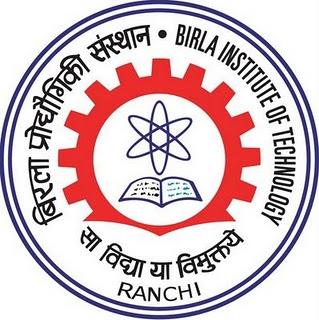
**Submitted by:**

**Prem Kamal (BE-CSE/1130/2011)**

**Saurabh Garg (BE-CSE/1146/2011)**

**Rahul Nori (BE-CSE/1159/2011)**

**( Group No – 29 )**



**Under the guidance of:**

**Dr. Sujan Kumar Saha**

**Assistant Professor**

**BIT Mesra**

**Date of Submission: 10th December 2014**

**ABSTRACT**

Online doctor is a term that emerged during 2000s, used by both the media academics, to describe a generation of physicians and health practitioners who deliver healthcare, including drug prescription, over the internet.

The patients describe their problem in the form of a paragraph or a detailed passage on the website. Online doctors go through these descriptions and provide a suitable solution.

This project focuses on summarizing the results of a blog written on the online medical websites in a few words, to detect the possible disease or a set of possible diseases. Some of the benefits of this project are that the patient receives a quicker reply and time spent on reading descriptions by doctors is reduced. It will also ensure that a doctor of a specific field does not end up spending his precious time on reading descriptions of problems of another genre. Consequently, doctors won’t refrain from replying in case of large sized written blogs.

**ACKNOWLEDGEMENTS**

Firstly**,** we would like to express our sincere gratitude to our project supervisor Dr. Sujan Kumar Saha, Assistant Professor, BIT Mesra, who guided us, all through the project and showed us the right path. We are very thankful that he has spent so much time with us during my project work.

My honourable thanks to Dr. Sandip Dutta, Associate Professor and Head of Department of CSE, BIT Mesra for providing excellent labs and library.

We thank BIT Mesra for providing us with necessary infrastructure and technical support that was required for this project.

Finally, we express our gratitude to our parents and friends for their valuable suggestions and moral support.

Prem Kamal (BE/1130/2011)

Saurabh Garg (BE/1146/2011)

Rahul Nori (BE/1159/2011)

BIT Mesra

**CONTENTS**

1. **Introduction**
   1. Online Doctoring
   2. Current Scenario
   3. Problems associated with Online Doctoring
   4. Motivation and Objectives
2. **Literature Survey**
3. **Data Collection**
   1. Dictionary of words
   2. List of symptoms
   3. Sample data
4. **Algorithms**
   1. Spell Checking Algorithm
   2. Algorithm to find the root word of all the words in the description
   3. Extracting all the symptoms from the input
5. **Design and Implementations**
   1. Dictionary Extractor
   2. Symptoms List Extractor
   3. Spell Checker ( Code cleaner )
   4. Root Word Converter
   5. Symptoms Extractor from Input

**Conclusions and Future Work**

**References**

**Chapter 1: Introduction**

* 1. **Online Doctoring**

**Online doctor** is a term that emerged during the 2000s, used by both the media and academics, to describe a generation of physicians and health practitioners who deliver healthcare, including drug prescription, over the internet.

Health advice is now the second-most popular topic that people search for on the internet. With the advent of broadband and videoconferencing, many individuals have turned to online doctors to receive online consultations and purchase prescription drugs.

Advantages of Online Doctoring –

1. Cost savings and convenient.
2. Accessibility and improved privacy and communication.
3. Patients can consult licensed physicians online – from anywhere, at any moment.
4. It allows doctors to give immediate online health tips and advice in rural areas through the internet broadband and videoconferencing system.
   1. **Current Scenario**

There are hundreds of medical websites that offer the patients online medical facilities. Examples include:

1. https://www.doctorspring.com
2. https://www.icliniccare.com
3. https://www.icliniq.com

The patients describe their problem in the form of a paragraph or a detailed passage on the website. This can be done 24 hours a day, 7 days a week, be it any remote location. Also, the patients can attach the required prior diagnosis from other doctors. There are many online doctors who go through the descriptions and provide a solution in their free time. These doctors who are available on the online websites are certified in their field of expertise.

* 1. **Problems associated with online doctoring**

Most of the problems associated with online doctoring are a result of the descriptions given by the patients. Some of the problems are:

1. The doctors are usually busy attending to their offline clients and cannot spend much time reading the lengthy descriptions.
2. The patient describing his/her problem may not be proficient with the language in which they are posting the description and hence there may be many errors. The doctors, being busy, may not be able to spend much time deciphering or correcting the mistakes.
3. Many a times, a doctor ends up reading the complete description and then realizes that the disease the patient is describing is not of his genre which results in unnecessary usage of his/her time.
   1. **Motivation and Objectives**

The purpose of the project is to summarize the results of a written blog on the online medical websites in a few words, to detect the possible disease or a set of possible diseases.

This will lead to following advantages:

1. A quicker reply to the patient.
2. A doctor’s time spent on reading the whole description is reduced.
3. Doctors won’t refrain from replying in case of large sized written blogs.
4. A doctor of a specific field doesn’t end up spending his precious time on reading descriptions of problems of another genre.

Goal is to make a full fledged system for doctors to get a quick review of the problem description given by the patients on various online medical sites.

The main objectives to implement the goal are –

1. Create a database of the disease and its related symptoms.
2. Cleaning the description provided by the user.
3. Deriving the root words from the cleaned text.
4. Finding out the symptoms from the information derived.
5. Finding the probable disease(s) for the symptoms described.

**Chapter 2: Literature Survey**

The first medical consulting website in the US was WebMD, founded in 1996 by Jim Clark (one of the founders of Netscape). Currently, its website carries information regarding health and health care, including a symptom checklist, pharmacy information, drug information, blogs of physicians with specific topics, and a place to store personal medical information.

In the UK, e-med was the first online health site to offer both a diagnosis and prescriptions to patients over the Internet. It was established in March 2000 by Dr. Julian Eden.

The project requires the study of stemming algorithms and spell checking techniques.

The first published stemmer was written by Julie Beth Lovins in 1968.

A later stemmer was written by Martin Porter and was published in the July 1980 issue of the journal *Program*. This stemmer was very widely used and became the de facto standard algorithm used for English stemming.

Ralph Gorin, a graduate student under Earnest at the time, created the first true spelling checker program written as an applications program (rather than research) for general English text: Spell for the DEC PDP-10 at Stanford University's Artificial Intelligence Laboratory, in February 1971.  Gorin wrote SPELL in assembly language, for faster action; he made the first spelling corrector by searching the word list for plausible correct spellings that differ by a single letter or adjacent letter transpositions and presenting them to the user.

In this chapter, literature survey on spell checking techniques and stemming algorithms is presented.

1. Spell Checking Techniques in NLP: A Survey – International Journal of Advanced Research in Computer Science and Software Engineering, Neha Gupta, Pratishtha Mathur, Volume 2, Issue 12, December 2012.

1. Explained approaches and roles of spell checkers in various applications.  
2. Explained the types of errors and error detection and correction methods.

1. A Comparison of Standard Spell Checking Algorithms and a Novel Binary Neural Approach – Victoria J. Hodge and Jim Austin – Volume 15 – No. 5 – 2003 – IEEE Transactions on Knowledge and Data Engineering.  
     
   1. Compared the standard spell checking algorithms.
2. A Comparative Study of Stemming Algorithms – Anjali Ganesh Jivani et al, Volume 2(6), International Journal of Computer Technology and Applications.  
     
   1. Explained stemming and lemmatization.  
   2. Explained different methods of stemming and their comparisons.

**Chapter 3: Data Collection**

**3.1 Dictionary of words**  :

A list of words is created to implement the spell checker program.

The words are extracted from the website –

http://www-personal.umich.edu/~jlawler/wordlist

The total number of words in the list is 69,903.

Technologies used in extraction are JAVA (Regular Expressions, File Read/Write Modules and URL package)

Algorithm –

Regex matching to find the pattern in the source code of the website.

**3.2 List of symptoms** :

Database consists of all the symptoms which is needed to extract information from the input given by the patient.

The required information is extracted from the following websites –

1. http://www.medicinenet.com/
2. http://symptomchecker.webmd.com

Technologies used are – Python 2.7, urllib, urllib2, bs4

Algorithm –

The pattern is found out in the page source of the website and the contents are derived by separating the tags.

**3.3 Sample Data**

The inputs given by the patients can be extracted from the websites:

1. https://onlinedoctor.lloydspharmacy.com/
2. https://www.lloydsonlinedoctor.ie/
3. http://www.netdoctor.co.uk

Technologies Used – Python

Algorithm –

Cron Job / Timed requests by the doctor.

**Chapter 4: Algorithms**

**4.1 Spell Checking Algorithm –**

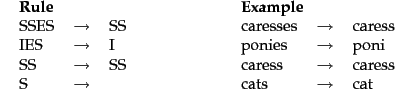
The misspelt or erratic words are to be replaced with the most suitable word in the context. Statistical based approach can be used to find the most suitable word for replacement. The list of all English words created previously will be used for this purpose.

***Algorithm –***

1. If the word is present in the dictionary, directly return the word.
2. If it is not, then choose the one that requires the least number of changes to form the possible word. The changes include: insertion / deletion / replacement / transposition of characters and splitting of word, all given equal weights.
3. In case of ambiguity, the terms that are medically more suitable are printed out as they have the maximum candidature value.

**4.2** **Algorithm to find the root word of all the words in the description –**

Porter’s algorithm is used to find out the suitable root word by eliminating the possible verb forms. It follows the following rules. Once a word has been form, we find it in the dictionary to check if it exists or the closest word that exists for its replacement.

****

**4.3 Extracting all the symptoms from the input –**  
 Algorithm–

1. Check every root word from the description if it is available in the database of symptoms. This check is made only when the word has not already been marked.
2. If such a word is found, it is marked and added to symptoms set.
3. A map is used to ensure that a single symptom is not marked and counted more than once.

Technologies Used – Python + List of symptoms created previously

**Chapter 5: Design and Implementation**

The design and implementation of the important code snippets are presented below:

1. **Dictionary Extractor**

**Design:**

Using regular expression matching, the words have been scraped out of the online dictionary file

**Code Snippet:**

***import java.io.BufferedReader;***

***import java.io.IOException;***

***import java.io.InputStreamReader;***

***import java.net.HttpURLConnection;***

***import java.net.MalformedURLException;***

***import java.net.URL;***

***import java.util.regex.\*;***

***import java.io.BufferedWriter;***

***import java.io.File;***

***import java.io.FileWriter;***

***import java.io.IOException;***

***public class Dic***

***{***

***public static void main(String[] args) throws IOException***

***{***

***String inputLine,s="";***

***int a[]={19,9,21,10,10,9,8,10,9,1,2,10,15,8,6,23,1,9,21,12,4,7,2,1,1,1};***

***Pattern patt = Pattern.compile("(<td><li><a href=.\*[0-9]\">)(.\*)(</a></li></td>)");***

***File file = new File("C:\\lab\\fms\\Dictionary.txt");***

***FileWriter fw = new FileWriter(file.getAbsoluteFile());***

***BufferedWriter bw = new BufferedWriter(fw);***

***for(int i=97;i<=122;i++)***

***{***

***String p="http://www.medilexicon.com/medicaldictionary.php?l=";***

***char q=(char)i;***

***String h= p+q;***

***URL x = new URL(h);***

***BufferedReader in = new BufferedReader(new InputStreamReader(x.openStream()));***

***while ((inputLine = in.readLine()) != null)***

***{***

***Matcher m = patt.matcher(inputLine);***

***while (m.find()) {***

***bw.write(m.group(2)+"\n");***

***}***

***}***

***in.close();***

***if(a[i-97]>1)***

***{***

***for(int j=2;j<=a[i-97];j++)***

***{***

***p="http://www.medilexicon.com/medicaldictionary.php?l=";***

***q=(char)i;***

***h= p+q+"&s=&p="+Integer.toString(j);***

***URL x2 = new URL(h);***

***BufferedReader in2 = new BufferedReader(new InputStreamReader(x2.openStream()));***

***while ((inputLine = in2.readLine()) != null)***

***{***

***Matcher m2 = patt.matcher(inputLine);***

***while (m2.find()) {***

***bw.write(m2.group(2)+"\n");***

***}***

***}***

***}***

***in.close();***

***}***

***}***

***bw.close();***

***}***

***}***

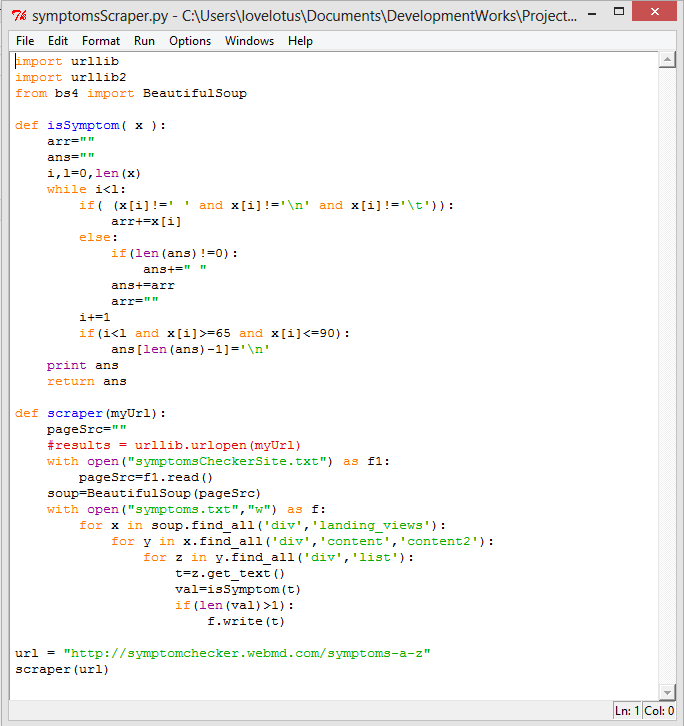
**( Coded in Java )**

1. **Symptoms List Extractor**

**Design:**

The symptoms have been extracted from the ‘symptomschecker.net’ website using a Beautiful Soup module, which lets the developer identify the pattern of the different tags and enables the user to find and extract the suitable data.

**Code Snippet:**

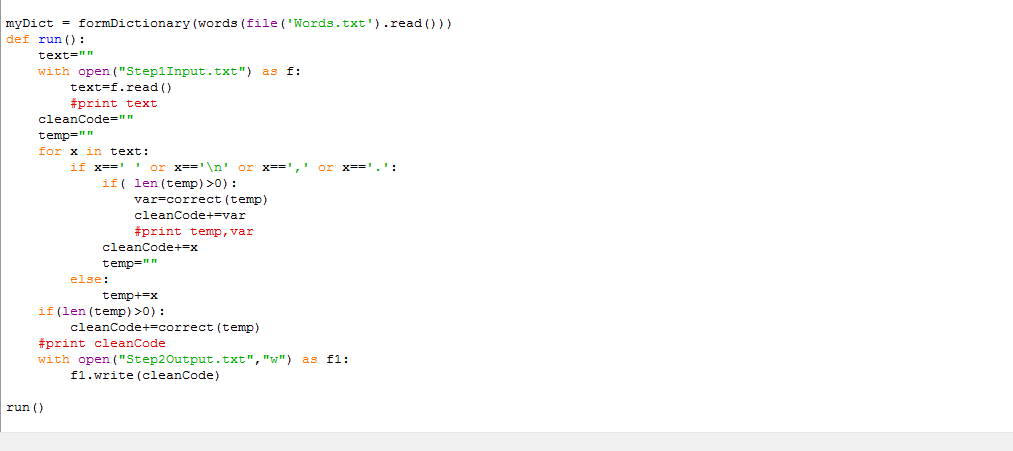
****

1. **Spell Checker ( Code cleaner )**

**Design:**

The spell checker algorithm uses a predefined dictionary, with the correct words as input and increased frequency of the words, those are more likely to be present in form of symptoms. The algorithm has been described already in section 4.1

**Code Snippet:**

****

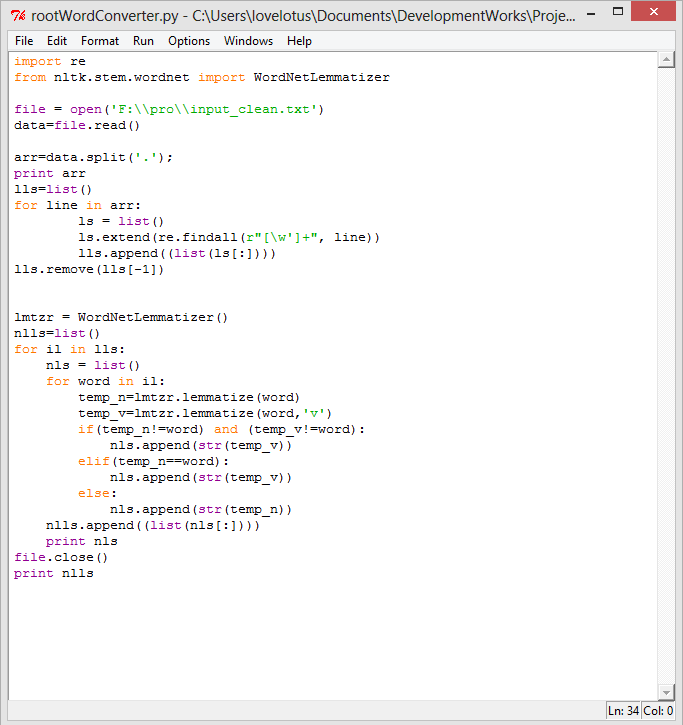
1. **Root Word Converter**

**Design:**

The first step is reduction to words according to the Portar’s algorithm

The next step is to lemmatize the words to find the word that is closest to it in the dictionary.

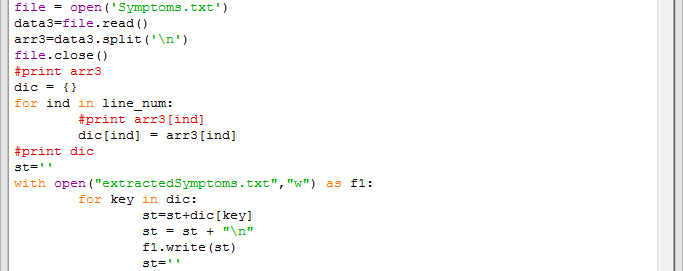
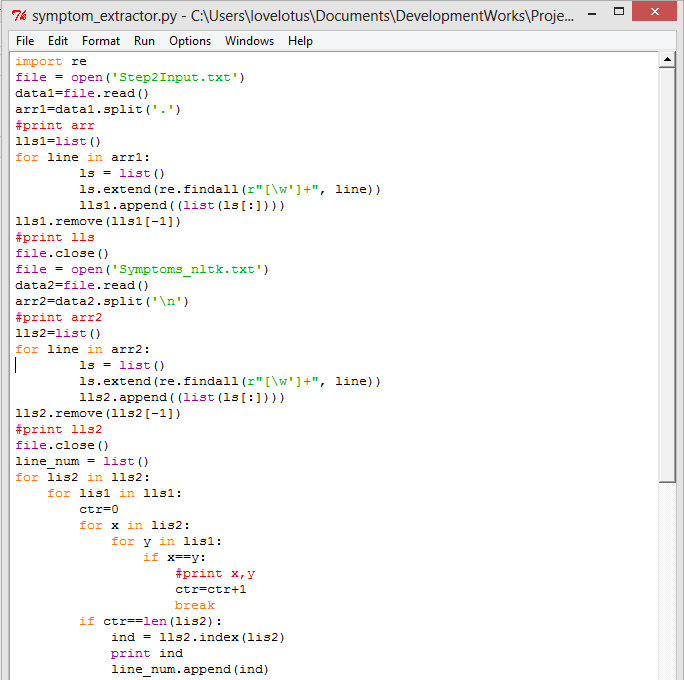
**Code Snippet:**

****

1. **Symptoms Extractor from Input**

**Design :**

Each symptom is searched in the complete input file for its presence. If the presence is found, the symptom is mapped, to ensure that it is not repeated again.

**Code Snippet :Conclusions and Future Work**

Firstly, online doctoring, its advantages and disadvantages and the current scenario are discussed. Then, the problems faced by the online doctors are explained in detail.

Secondly, the process of how the databases (used in different algorithms) have been constructed has been explained.

Finally, an algorithm has been developed to create a list of symptoms from the input given by the patients. For the achievement of this objective, firstly, spell checking algorithm has been developed and then a modified version of the Porter’s algorithm has been used to find the best possible root word.

In the future, the possible disease or a set of possible diseases based on the symptoms extracted shall be detected using machine learning concepts or fuzzy logic.

**References**

* Spell Checking Techniques in NLP: A Survey – International Journal of Advanced Research in Computer Science and Software Engineering, Neha Gupta, Pratishtha Mathur, Volume 2, Issue 12, December 2012.
* A Comparison of Standard Spell Checking Algorithms and a Novel Binary Neural Approach – Victoria J. Hodge and Jim Austin – Volume 15 – No. 5 – 2003 – IEEE Transactions on Knowledge and Data Engineering.
* Spell Checking – http://en.wikipedia.org/wiki/Spell\_checker
* Stemming - http://en.wikipedia.org/wiki/Stemming
* Porter’s Algorithm - http://snowball.tartarus.org/algorithms/porter/stemmer.html
* A Comparative Study of Stemming Algorithms – Anjali Ganesh Jivani et al, Volume 2(6), International Journal of Computer Technology and Applications.