HEALTH CARE DATA MINING

Dhiren Tejwani, Kaushik Sampath, Rahul Arora, Raj Buddhadev, Riddhi Patel, Rishab Banerjee, Tithi Patel

1 ABSTRACT

Nowadays people surf medical forums and websites to look for the symptoms, diseases and read about the first hand experiences of these symptoms from other patients. Diabetes is one of the most common and widespread diseases across the globe. This paper deals with the machine learning and information extraction techniques for identifying the knowledge relevant to Diabetes and its various types from common healthcare websites and blogs.

Keywords: *Diabetes, Symptoms, SVM, CRF, symptom-symptom relation graph, UMLS Ontology, Levenshtein distance*

2 INTRODUCTION

The healthcare domain gives various opportunities for retrieving information from heterogeneous sources, transfer them into structured form and apply them into some application.

One of the biggest challenges in healthcare systems is the explosive growth of the data. Web Mining techniques like Information extraction, Ontology implementation, and Intelligent searching are useful to extract meaningful knowledge from the large dataset. Our project aims to show relevant symptoms for different types of diabetes and find the discussion threads for a searched type or symptom and sorts them by maximum relevance. Sites like WebMD and diabetes.co.uk provides the dataset (discussion threads) that is needed for the implementation of the project. Since there are several available forums which offer discussion on many diseases, the project tries to provide a filter-based interface, where a user can conveniently look for discussions.

3 PROBLEM STATEMENT

There have been research works that focus on mining healthcare data from discussion forums and apply various machine learning techniques on it to generate some helpful information. However, there are not many platforms available that allows users to find all the relevant information about a particular symptom or disease at one place. To address this issue, we came up with a system that allows the user to browse through various discussion threads mined from discussion forums and search discussions related to the symptoms and disease they are suffering from. We also provide users with suggestions of related symptoms to their searched symptom.

In future, we plan to further enhance our system by suggesting users the medications that they can take to treat/cure a symptom or disease. Currently, our system provides discussion threads related to diabetes only, but we plan to expand our system for various diseases.

4 RELATED WORKS

In the last two decades, a variety of different ML techniques and feature selection algorithms have been widely applied to disease prognosis and prediction. The vast majority of publications makes use of one or more ML algorithms and integrates data from heterogeneous sources for the detection of tumours as well as for the prediction/prognosis of a cancer type. The state-of-the-art areas include Breast Cancer Detection, Lung Cancer Diagnosis, Skin Cancer Classification and the like.

There have been many interesting works in the domain of healthcare that uses the power of data to provide a useful analysis of symptoms, diseases, and its detection. However, there isn't much recognized work done when it comes to discussion forums in the

healthcare domain apart from the professional healthcare forums. There have been few ideas such as [1] and [3], where the authors have mined the data from discussion forums and have performed various machine learning tasks to find relationship between symptoms. But, we provide an extension of these ideas by building a user interface that allows a user to browse through various discussion threads mined from discussion forums. We also provide functionalities like searching, filtering and common symptoms that enhances the user experience and provides a specific solution to the user's search.

5 SYSTEM ARCHITECTURE AND ALGORITHMS

The system that we have built revolves mainly around the data that we have collected from discussion forum and the algorithms that we have applied on the raw data to get the desired results. Our system is divided into two different parts: information extraction and system architecture. Figure 1 helps to explain the information extraction process and data preparation methodology.

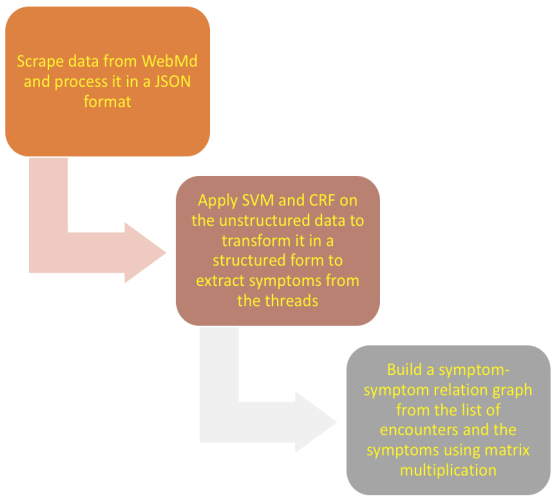

Figure 1: Information Extraction Methodology

5.1 Finding symptoms using CRF and SVM

To convert unstructured data scraped from the WebMD into structured data, we use algorithms like CRF and SVM. CRF helps to classify unstructured data into a sequential structured data. We classify unstructured data into a structured format using tags <symptoms> and <medications>. This structured data contains sentences instead of words, hence we needed to extract symptoms out of the sentences listed in <symptoms> tag. For this, we use SVM to classify key symptoms with the help of UMLS ontology framework. To implement these algorithms on medical forum data we referred to [1] and came up with similar implementation.

5.2 Symptoms Similarity Matrix

We have discussion threads in our project instead of clinical notes as suggested in [3]. There are many different types of encounters of discussion threads possible. Each thread contains a specific kind of information regarding the symptoms and diseases. Graph generation is required to find out the relationship between different types of symptoms. There is an edge between two symptoms with some weight if they are related.

Firstly, we make an encounter matrix N. All the rows of N correspond to symptoms and all the columns of N correspond to encounters.

Nij = 1, if symptom i is in encounter note j.
Nij = 0, otherwise.

Then we make transpose of that encounter matrix NT. Now, All the rows of N correspond to encounters and all the columns of N correspond to symptoms. We then do matrix multiplication of these matrices N and NT and we have a final matrix S. Matrix S is the symptom-symptom similarity matrix and values correspond to the weights. The symptoms extracted from the same message thread form a clique and such individual cliques are summed together to form the graph G with the edges having the weight w_{ij}.

We have taken into consideration the local sensitive construction while forming the symptom graph. When any two symptoms are encountered nearby (consecutively) in the same thread we assign weight which is higher as compared to the two symptoms which are encountered at a greater relative distance. Hence the weight of the edges are normalized using the distance between the symptoms in a thread.

Further the frequent symptoms are penalized in order to alleviate the problem where some symptoms are very common like fever. We use normalize the matrices formed by dividing the individual cells by the column sum. In this way we form the Markovian Matrices for each of the formulated graphs. The normalized graph G1, G2, G3… are combined as in [3] form the aggregated graph G by averaging the symptom matrices. To expand this graph with the help of the existing set of symptoms we use relevance score. For all the immediate symptoms in the graphs we do not worry as all the immediate edges are considered. For the other symptoms we use algorithm which is like PageRank algorithm. The Random Walk with Restart algorithm is a recursive algorithm and uses a damping factor.

$$r = cGr + (1 - c)\, e$$

Here the relevance score is r, the existing set of symptoms vector taken from the WebMD site, c is the probability with which one of the outgoing edges of the current symptom is selected and 1-c is the probability with which we are restarting the random walk again from the existing set of symptoms (the set which we are trying to expand). The restart probability for our graph expansion is taken as suggested by Google's PageRank algorithm. This helps us in controlling the range from the existing symptom the walker will explore. After the max iterations for calculating the relevance scores we take the five symptoms having the highest relevance score apart from the immediate symptoms.

5.3 Stemming

We implemented stemming for a purpose of normalizing words. Many words that have the same meaning and carries different tenses can be handled by stemming method. We maximized the discussion thread searching by implementing this process. It manipulates the words in the search filter and retrieves the root word.

5.4 System Architecture

The system is implemented as two tier web application with a Django Server at the backend and a Single Page HTML Web Application on the frontend utilizing Bootstrap 4.0 for effective presentation, jQuery for smart filtering and front end processing and highlighting.

When a user enters a keyword and searches, an ajax request is made by the frontend Web App to the backend Django Web Service. The GetThreads REST API, accepts a keyword and passes it to the GetRelatedSymtoms function. This function internally employs SVM and provides a list of related symptoms and corresponding threads from the symptom graph ordered by increasing edge weights from the node corresponding to the searched symptom. Then these main threads are merged with the corresponding reply threads available from the scraped , transformed data. Now the combined threads and replies for the searched symptom along with the mined related symptoms are returned as response to AJAX request made by the web app. Then the web app filters the returned threads further based on the Diabetes Type Filters selected by the user and presents the results to the user.

The related symptoms in decreasing order of relevance to the searched symptom or condition are presented to the user in the rightmost pane of the web application. The left pane displays the related threads containing the identified related symptoms and also highlights each occurrence of all related symptoms within the threads and replies for easy navigation for the user.

Figure 2 shows the user interface of our system. Figure 3 explains the overall flow of the system.

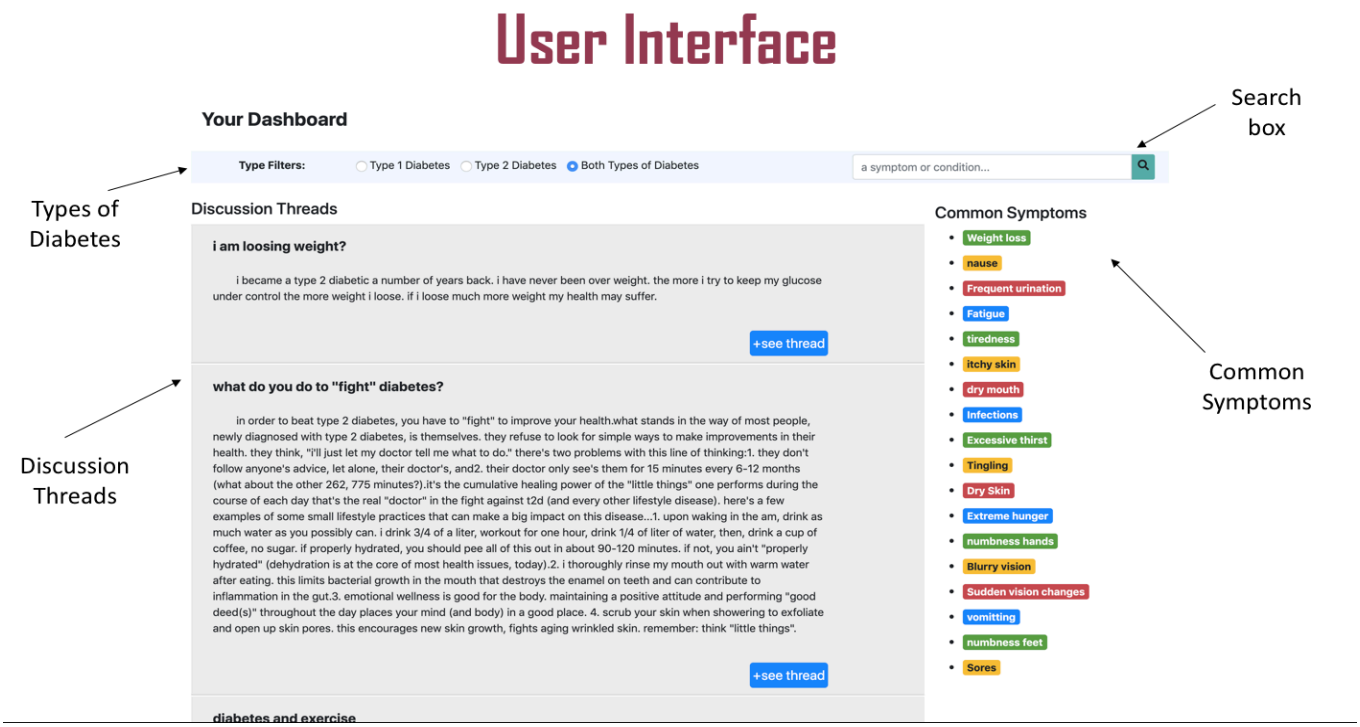

Figure 2: User Interface of the system

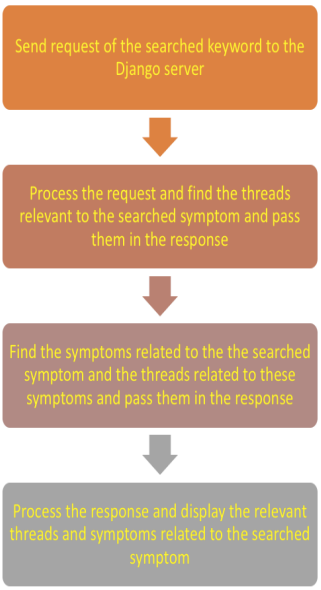

Figure 3: System Flow

6 DATASET

Web scraping also termed as Web Data Extraction is a technique employed to extract large amounts of data from websites whereby data is extracted and saved to a database. Data displayed by most websites can only be viewed using a web browser. They do not offer the functionality to save a copy of this data. To crawl multiple pages from the website, we have implemented web crawlers. We needed data from discussions. Discussions that are nested and sometimes follow a dynamic structure. After thorough research and looking through many different discussion forums, we decided to use the discussions from WebMD. WebMD is best known as a health information services website, which publishes content regarding health and health care topics. The reason why we chose this website is that we are focusing on Diabetes and it had different sections of discussions for different major diseases. The message boards contain a great content, a lot of different users, and hence this leads to significant data in each thread.

The structure of the message board is similar to a general discussion forum. There is the main page that

displays a list of all the threads. The list has the thread title and thread body. On clicking each thread title, the actual content page opens that contains the entire thread body along with its replies. One more reason to select this website is that the users can also reply to the replies of a thread. Hence it has a complex nested structure.

For scraping data, we have used the Scrapy framework implemented inside Python 3 scripts. So, the web scraping script starts with the urls of all the pages that contain threads related to diabetes. Once, the page is crawled, the scraper goes through each of the threads and fetches the link to the page content of it. The crawler then opens the page content of the thread. With this page loaded, the scraper gets the title, body, and all the content of the thread and its replies. This data is stored in the json format.

The above-fetched data is in raw format and needs to be cleaned and structured. This process includes escaping all the HTML tags, structuring the data in a thread-reply format and sorting the threads into its types. After these processes, the data is ready to be used for the further operations of the system.

7 EVALUATION

We found that SVM with the kernel trick gave better precision and recall values when compared to CRF based classification of sentences into symptoms and medications. Simple SVM however, didn't perform well-- the f1 scores we obtained using this simple classifier was very low, leading us to conclude that the data is not linearly separable. This served as the motivation to use kernels. We tried gaussian, polynomial and sigmoid kernel, sigmoid kernel outperforming the others.

The common symptoms were found from the tags associated with the word Diabetes on the website, WebMD. These symptoms were processed on the entire data set to achieve a symptom similarity matrix, that evaluated the likelihood of another symptom based on some other symptom. This symptom similarity matrix, which is used in further stages, is one of the most important findings of the project.

Another important result of the project was mapping the threads with a symptom and ordering the threads based on the relevance from most important to least for all the relevant symptoms taken from the source, WebMD. This evaluated order is further passed on and displayed at the front end.

The implementation of search enabled us to carry out another significant experiment. The searched query was mapped to the symptoms set using Levenshtein distance. This enabled us to estimate the symptom which the user is looking for, in case the spelling of the word is a bit different from the word in the set, or the user misspelled the word, or used a similar word. Searching based on similar words and not just a few specific words, enabled us to add flexibility to the project, thereby not putting any kind of functional limits.

8 CONTRIBUTION

NAME	CONTRIBUTION
Dhiren	Data Scraping and Cleaning
Kaushik	Developed User Interface
Rahul	Integrated processed data with user interface
Raj	Implemented SVM & CRF on the unstructured data, developed backend system using Django server
Riddhi	implemented symptom-symptom related graph to find relationship between symptoms
Rishab	Implemented SVM & CRF on the unstructured data, developed backend system using Django server
Tithi	implemented symptom-symptom related graph to find relationship between symptoms

9 CONCLUSION

The built system is very capable of identifying related symptoms for a given symptom of diabetes with acceptable relevance. The designed system is extensible to cover more diseases will be helpful for medical industry at large. By leveraging multiple sub-domains of Semantic Web Mining like data-scraping, information extraction, data-transformation, machine learning, and presenting and reporting using powerful user interface, the built system is capable of helping patients and medical practitioners to learn more about

symptoms of a given disease by mining the useful information from discussions of existing patients from important venues on the internet like discussion forums. In future, the data extraction and transformation task from discussion forums can be automated to make the system more dynamic and capable of pulling up-to-date discussion threads and to discover new symptoms for a disease like diabetes. Also, the system can be extended to support multiple diseases. Such a system will be very fruitful to medical researchers trying to understand co-occurring symptoms for diseases which are not so well understood yet.

10 REFERENCES

[1] Parikshit Sondhi, Manish Gupta, ChengXiang Zhai, Julia Hockenmaier: Shallow Information Extraction from Medical Forum Data.

[2] Ronen Feldman, Moshe Fresko, Jacob Goldenberg, Oded Netzer, Lyle Ungar: Using Text Mining to Analyze User Forums.

[3] Parikshit Sondhi, Jimeng Sun, Hanghang Tong, ChengXiang Zhai: SympGraph: a framework for mining clinical notes through symptom relation graphs.

[4] Mamatha Balipa, Dr. Balasubramani R: Disease-Treatment Relationship Extraction for Psoriasis from Online Healthcare Forums using NLP and Classification Techniques.

[5] Yunzhong Liu, Yi Chen, Jiliang Tang, Huan Liu: Context-Aware Experience Extraction from Online Health Forums .

[6] Ilayaraja .M, Thiru Meyyappan: Mining medical data to identify frequent diseases using Apriori algorithm.

[7] Dr. Varun Kumar, MD. Ezaz Ahmed: An Empirical Study of the Applications of Web Mining Techniques in Health Care.

[8] Discussion Forum Source - WebMD (https://www.webmd.com).