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

Md. Tahmid Rahman Laskar
Department of CSE
Islamic University of Technology
Gazipur, Dhaka, Bangladesh

Md. Tahmid Hossain
Department of CSE
Islamic University of Technology
Gazipur, Dhaka, Bangladesh

Abu Raihan Mostofa Kamal
Associate Professor,
Department of CSE
Islamic University of Technology
Gazipur, Dhaka, Bangladesh

Nafiul Rashid
Lecturer,
Department of CSE
Islamic University of Technology
Gazipur, Dhaka, Bangladesh

ABSTRACT
Rapid proliferation of Internet technology and handheld devices has opened up new avenues for online healthcare system. There are instances where online medical help or healthcare advice is easier or faster to grasp than real world help. People often feel reluctant to go to hospital or physician on minor symptoms. However, in many cases, these minor symptoms may trigger major health hazards. As online health advice is easily reachable, it can be a great head start for users. Moreover, existing online health care systems suffer from lack of reliability and accuracy. Herein, we propose an automated disease prediction system (ADPS) that relies on guided (to be described later) user input. The system takes input from the user and provides a list (topmost diseases have greater likelihood of occurrence) of probable diseases. The accuracy of ADPS has been evaluated extensively. It ensured an average of 14.35% higher accuracy in comparison with the existing solution.

General Terms
Sentiment Analysis, Opinion Mining, Data Mining, Natural Language Processing.

Keywords
Relevant Attribute (RA) Data Structure, Word Tagging, Synonym Parent Tree, Reference Tag, Decision Tree.

1. INTRODUCTION
Number of internet users is growing exponentially over the years. In a national survey conducted by the Pew Internet Project [1] 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 [2], 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 [3].

However, predicting disease based on user (patient) input is a complete different ball game ([4], [5] and [6]). Generally, people express their symptoms in non-technical or natural terms which adds complexity in predicting diseases. In this work, the objective is to construct a novel architecture consisting of techniques that will allow disease prediction with greater accuracy based on user input. This paper is organized as follows.

Section 2 sheds light on related works. Contribution of this work is described in section 3. Section 4 contains overview of the model. Special data structure and algorithm of this architecture are introduced in section 5, 6. Section 7 contains probability computation procedure. Experimental results are shown in section 8. Finally, section 9 concludes the paper.

2. RELATED WORK
The work presented in [3] 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 text input with system database has much more complexity.

[4] 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 has been put on to predict specific diseases [7], [8]. For instance, authors in [7] focus 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. [9] 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 ac-count suggestions provided by different users on comment sections in disease analysis.

Healthcare websites such as isabelhealthcare.com, mayo-clinic.org, patient.co.uk, are providing disease prediction based on user input ([5], [10] and [6]). [10] uses jargon-laden interface (I.e. users need to navigate through a longlist 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. [5], [6] take guided input from user. However, they rely on mere symptom-disease relationship framework ([11], [12]) and use full text database [23]. 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.

3. CONTRIBUTION

In this paper, the contribution includes proposing a new disease prediction framework (ADPS) that takes into account symptom names as well as other vital parameters (to be described in section 5) to improve disease prediction accuracy and proposing techniques (to be described in section 5) to allow greater linguistic diversity so that users do not feel uncomfortable while giving input.

4. OVERVIEW OF THE MODEL

It is presumed 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.

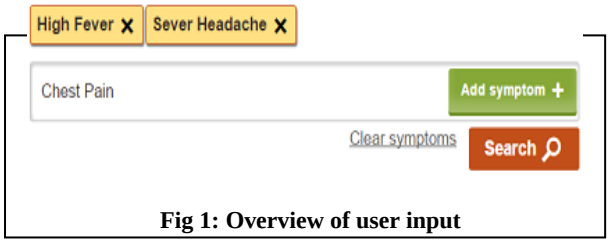

Fig 1: Overview of user input

5. ADPS COMPONENTS

5.1 Relevant Attribute (RA) Data Structure

Most of the existing disease prediction systems ([4], [5], [6]) 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. For example, 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 (High, Low etc.)

O = Organ name (Abdomen, Head, Heart etc.)

D = Duration (10 days, 1 month etc.).

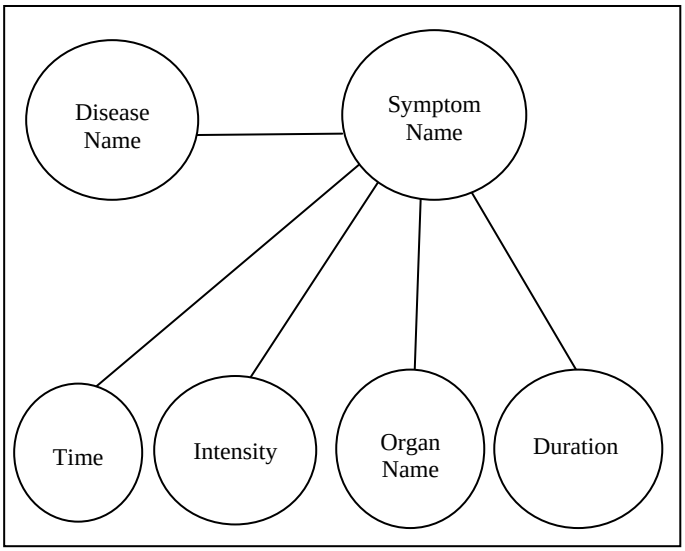

Fig 2: RA Data Structure

5.2 Disease Symptom Database

It is a disease symptom database developed from expert sources ([10], [15], [16]) 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	×	High	×	×
Pain	×	×	Eyes	×
Pain	×	High	Joint	×
Pain	×	×	Muscle	×
Fatigue	×	×	×	×
Nausea	×	×	×	×
Vomiting	×	×	×	×
Rash	×	×	×	×

Fig 3: DB representation for Dengue (Matrix D-D)

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

Fig 4: DB representation for Typhoid (Matrix D-T)

6. 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.

Tagging will be done using following three techniques:

i. Synonym Parent Tree

ii. Symptom Reference Tag and Decision Tree

iii. Relevant Attribute Array

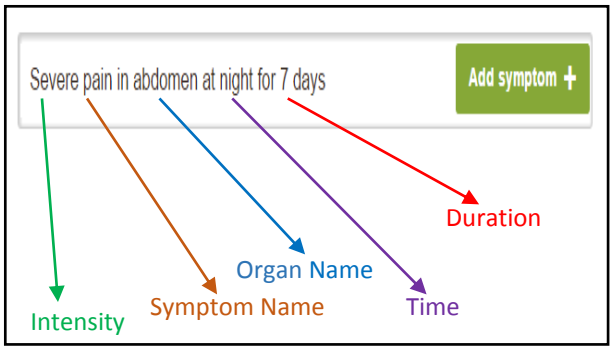

Fig 5: Word Tagging

6.1 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 with the 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, Synonym Parent Tree has been introduced. 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.

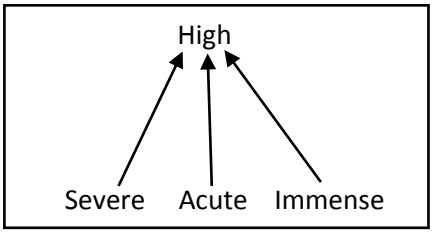

Fig 6: Synonym Parent Tree

6.2 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 fulfil 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 However, the above mentioned approach should still be able to interpret it as 'insomnia' even though the exact user input is not part of the database.

To realize the above mentioned scenario, a decision tree based solution is proposed to determine the symptom name from such compound inputs. To use the decision tree, a symptom associated tag is introduced. It is called '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.

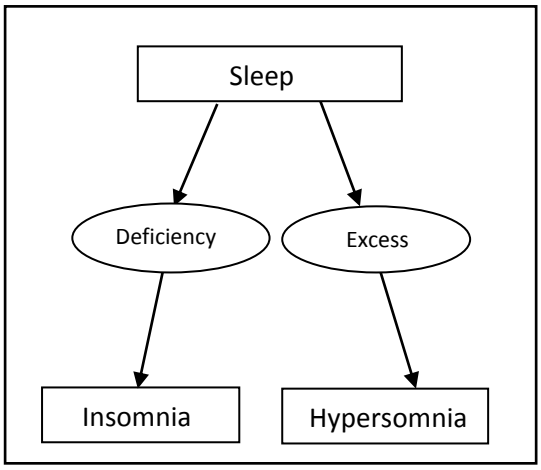

Fig 7: Decision Tree

An example of symptom reference tag with the word 'sleep' is given in Table 1.

Table 1. Reference tag example

Symptom Name	Reference Tag
Insomnia	Sleep
Hypersomnia	Sleep

6.3 Relevant Attribute (RA) Array

Once the type of each of the input words is determined using techniques described in section 5 and 6, words will be put on to five different arrays which is 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.

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

and if intensity[n] denotes 'High' intensity, it will refer to the symptom of the nth 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 can 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).

```
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
end function
```

Fig 8: Algorithm for tagging words

7. 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	×	×	×	×
Fatigue	×	×	×	×

Fig 9: Matrix Dq

Initially symptoms from this data matrix are retrieved and mapped with the symptoms in the database. Then data matrices corresponding to all diseases are recorded for further processing.

In this case, matrices in figure 3 and 4 are retrieved from database named D-D and D-T (See section 5.2).

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) ---------- (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 ([10], [15], [16]), 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, the empty rows are considered 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.

8. EVALUATION AND ACCURACY

For evaluation, visual studio 2015 is used as the platform. C# is used as the programming language and Oracle database is used to store the data. As stated before, ADPS provides disease predictions in ascending order like other existing systems.

Probability of each disease is divided in 2 groups. If the probability is between 1 to 50% (inclusive), the probability is considered to be as Low (L) and if the probability is between 51 to 100% (inclusive), the probability is considered to be as High (H).

To compare their relative accuracy, each of the ranked predictions is checked against the ground truth. The ground truth symptom-disease associations are recorded from [10], [15] and [16]. 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 low probability group, 4 in high. Considering it as ground truth, let us take a look at the following table:

Table 2. Ground Truth Comparison Table

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

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 groups is of degree 1, then it is considered as complete mismatch (0).

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%.

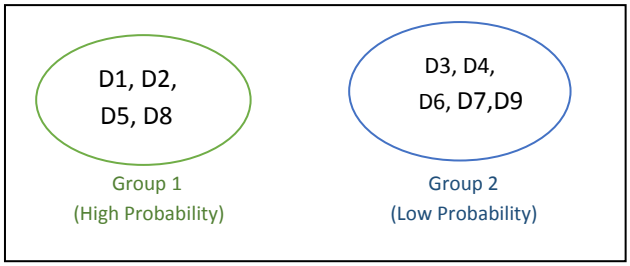

Fig 10: 2 Groups based on probability

This accuracy value resembles the quality of a predicted disease ranking list by a system (higher value means more accurate and lower value means less 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 the proposed approach, user queries [17] have been picked and user posts have been collected from [18]. We then run 10 experiments.

The results produced by ADPS (Using equation I) and disease-symptom matching system [11] are then arranged in tabular form like Table 2 for each disease.

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

Table 3. Accuracy from 10 Experiments

Experiment	Accuracy in disease symptom matching system [17]	Accuracy using ADPS	Improvement
E1	69.82%	81.61%	16.88%
E2	78.95%	83.42%	5.66%
E3	71.48%	78.25%	9.47%
E4	51.27%	63.35%	23.56%
E5	64.67%	73.38%	13.47%
E6	69.56%	77.73%	11.75%
E7	58.72%	61.42%	4.60%
E8	76.13%	91.75%	20.52%
E9	65.27%	81.57%	25.97%
E10	65.74%	73.35%	11.58%

An average of 14.35% higher accuracy is observed after evaluation with a minimum of 4.60% and maximum of 25.97%. It is worth noting that ADPS accuracy is significantly better. Therefore, disease prediction is more accurate in ADPS.

9. CONCLUSION

Technology has ushered numerous ways to drive mankind towards a better world, a better life. Mankind will be better off if technology is blended into our lifestyle. People rely on technology to find out solutions for problems they cannot solve by themselves. Health related issues are one of those where automated help can greatly benefit the attention seeker as the person is getting the necessary information by just a few clicks.

In this work, we show that 'Automated Disease Prediction System' 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 to 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.

10. REFERENCES

[1] Pew Research centre health fact sheet : www.pewinternet.org/fact-sheets/health-fact-sheet.

[2] http://edition.cnn.com/2012/05/04/tech/socialmedia/ facebook-lies-privacy [Accessed 07/06/2015]

[3] 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.

[4] 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).

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

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

[7] 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.

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

[9] 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).

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

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

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

[13] http://www.kiranreddys.com/articles/clinicaldiagnosis supportsystems.pdf [Accessed 30/10/2015]

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

[15] 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.

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

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

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

[19] 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.

[20] 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.