Abstract-- In this paper, we present generic architecture for building domain specific knowledge bases that can be automatically acquired by analyzing big data collected from the web environment. As a reference implementation, Nursing Home Application is developed based on our proposed architecture. The initial experiment result shows valuable warrant for future studies.

Keywords: Big Data Analysis, Automatic Knowledge Acquisition, Web Data Mining, Internet Programming

1. Introduction

From the inception of web space, endless resources of information for users to search for every day are available in the web space, and users seek intelligent search for effectively finding useful information for their daily need. For example, given some medical symptoms, users want to search a list of medical specialists in a specific geographic region. Or users want to find restaurants in a specific cuisine that meet their need in terms of quality, prices, atmosphere, and so forth. In other words, users become more and more interested in getting effective answers from the web as a Knowledge Based System (KBS).

Traditionally, a KBS often termed as an Expert System [1], makes extensive use of specialized domain knowledge to solve problems at the level of human expert. The knowledge base used in such a KBS may be either expertise elicited from human domain expert(s) or knowledge which can be extracted from resources such as professional books or magazines.

In general, as shown in the Figure 1.1, a KBS contains the knowledge base with which the inference engine of the KBS makes a conclusion. These conclusions made by the KBS are answers with respect to the user’s query. The answers are formulated as facts or a list of potentially relevant suggestions.

The KBS is generally designed for a specific problem domain. The conventional way of building a knowledge base is carried out by having knowledge engineers repeat the cycle of interviewing the domain experts, constructing a prototype, testing, and interviewing again. Such knowledge acquisition process is very time consuming and labor intensive task. To extract domain knowledge from a big data set often available in the web space is an alternative way to get knowledge from domain experts.

To reduce knowledge acquisition effort, knowledge bases used in KBS are often formulated as heuristics [1]. Heuristics are termed as shallow knowledge or empirical knowledge gained from experience which may help producing potentially useful solutions.

One of the most important and difficult knowledge acquisition tasks is to generalize the target domain to formulate (or discover) its primary concepts (or conceptual categories), and terms (or attributes) that describe a concept, and the relationships between concepts and terms. Knowledge acquisition task should involve a process for identifying important concepts in a specific domain and involve a process of determining the degree of relationships between concepts and terms. Such relationships are heuristics that are the basis of the KBS.

In this paper, we are motivated to develop a generic architecture which automatically builds a domain specific knowledge base by analyzing a big data set collected from the web space. A list of concepts and terms (attributes) that describe concepts are assumed to be given or extracted from a set of web sites in a specific domain.

The proposed method is based on statistical feature of concept and term co-occurrence in a specific domain. If a term co-occurs with a concept in many web documents, the term is considered more relevant to the concept. Such heuristics are basis for our knowledge base.
The concepts in a specific domain in the proposed architecture can be manually derived or extracted from the semantic web based Resource Description Framework (RDF) in a specific domain. For example, as a case study, we developed a prototype which is used to search medical specialties, for nursing home application, based on the knowledge base (heuristics) acquired automatically by the proposed knowledge acquisition system. The prototype is based on the concepts (i.e., specialties) and terms (i.e., symptoms) extracted from the RDF of well-known medical institutions [4, 5].

The salient features of the proposed architecture are summarized below:

1. Knowledge base in a specific domain is automatically acquired by analyzing term co-occurrence between a set of concepts and a set of terms. Term co-occurrence frequencies are obtained by a simple web crawler that makes use of existing search engines such as Bing and Altavista.

2. Knowledge acquisition of the proposed architecture is generic in the sense that the weight (or degree of relationship) between concept and a term can be automatically calculated in any domain. Term frequency and inverse term frequency are used to calculate the degree of relationship more accurately for effective query processing.

3. Knowledge representation is based on simple inverted file, which can be formulated as normalized tables of a simple database schema in a DBMS.

4. User query is a simple vector whose element represents the importance (normalized weight) of a term in the query.

5. Inference engine is based on the cosine similarity measure between user query vector and a concept vector. Inference engine (i.e., query processor) is thus time efficient because of simple query and knowledge representations.

In section 2, automatic knowledge acquisition along with overview of the proposed architecture will be explained. In section 3, two different algorithms to extract knowledge base from the collected data set are presented. An algorithm for implementing query processor as an inference engine is also explained.

In section 4, we present the Nursing Home Application (NHA) as a reference implementation of the proposed generic knowledge based search architecture. In section 4, experiment results are shown for validating the utility of the proposed architecture based on NHA as a case study. Conclusion of the paper is presented in section 5.

2. Proposed Architecture and Knowledge Acquisition Framework

Knowledge base is formulated by first identifying the important terms and concepts, and then determining the interrelationships between concepts and terms in a specific domain.

Personal Construct Theory (PCT) has been frequently used to elicit concepts and terms in a specific domain from domain expert [6]. However, PCT based approach for eliciting concepts and terms often requires much longer turnaround time. In our case conceptualization processes are carried out by extracting concepts from well-defined RDF based data set in a specific domain.

Upon completing the conceptualization process, our web brokering architecture will run a web crawler which collects concepts and term co-occurrence statistics from a well-known search engines (such as Altavista, Bing, Yahoo, Google, etc..). As an example, our web crawler will give a query that consists of concept \( C_i \) and term \( t_j \) pair, and get the frequency of documents where \( C_i \) and \( t_j \) co-occurs from each search engine. This querying process continues until we exhaustively collect co-occurrence document frequency statistics with respect to all concept and term value pairs. The frequency statistics are collected from multiple search engines to get unbiased statistics (refer to the Figure 2.1). Inverse term frequency is also considered to calculate accurate relationship between a term and a concept.

Initially, co-occurrence frequency matrix between concepts and terms is formulated in our document frequency database. Multiple document frequency matrices are constructed for multiple search engines. The frequency will be analyzed to calculate importance of each term in a concept.

For example, in a specific medical domain, given a concept \( C_i \) (specialty) named “internal medicine”, and a set of terms (symptoms) such as “loss of weight”, “loss of appetite”, and “low blood pressure”.

Then the frequency vector corresponding to the concept \( C_i \) will be converted into a weight vector that can be represented as an inverted list, as shown in Figure 2.2. All these inverted lists of all the concepts will be generated and converted into the normalized database tables as the knowledge base of the proposed architecture.

Maintenance of concepts and terms are carried out by the administrator as shown in the Figure 2.1. Term frequency and inverse term frequency are used to calculate the degree of relationship between a concept and a term more accurately for effective query processing. In section 3, the use of inverse term frequency is illustrated when the degree of relationship is calculated.
3. Algorithms for Building Knowledge Base and Query Processor

In this section, we explain how to calculate the importance (weight) of a term $t_j$ in a concept $C_i$, which is the central component of our proposed knowledge acquisition system. We then present an inference algorithm that is the basis of the query processor.

3.1 Bucket Algorithm

Given a set of concepts and a set of terms that describe concepts, we start by checking the co-occurrence of both concept and terms from the web document frequency database we collected from the web (refer to the Section 2).

Our novel bucket algorithm can be explained by the following example: if a particular concept $C_i$ and term $t_j$ pair $(C_i, t_j)$ co-occurs in one million documents, while the concept $C_i$ and term $t_k$ $(C_i, t_k)$ co-occurs in one thousand web documents, then the correlation (or degree of relationship) of $(C_i, t_j)$ is stronger than $(C_i, t_k)$. We need to consider the co-occurrence frequencies of all $(C_i, t_j)$ pairs in the collected
frequency database, where $|C_i|$ is $n$ and $|t|$ is $m$. In other words, the number of concepts (i.e., conceptual categories) in a domain is $n$ and the number of terms that describe each concept $C_i$ is $m$.

There is a need to normalize the data so that multiple search engines use the same scale. For example, search engine Bing could return frequency values lying between 1 thousand and 1 billion, while Altavista could return frequencies between 500 and 500 million. The way to normalize document frequency data is the basis of the bucket algorithm, where min and max results found for any term $t_j$ of a particular concept $C_i$ and the possible values are divided in 10 buckets with the Equation 1 shown below. Let us define $f$, $max$, and $min$ as follows:

$f$: \{\{C_i \cap t_j\} | the number of web documents where $C_i$ and $t_j$ co-occurred\}

$max$: \{\{C_i \cap t_j\} | the maximum number of web documents where $C_i$ and $t_j$ co-occurred, where $j = 1$ to $m$ and $m$ is the number of terms\}

$min$: \{\{C_i \cap t_j\} | the minimum number of web documents where $C_i$ and $t_j$ co-occurred, where $j = 1$ to $m$ and $m$ is the number of terms\}

\[
W(C_i, t_j) = \text{trunc} \left( \frac{f - \text{min}}{\text{max} - \text{min}} \right) - 1
\] (1)

The equation 1 explains that by knowing the difference between the results found for a concept $C_i$ and a term $t_j$ minus the lower bound (min) found for the concept $C_i$, we can then divide by the range of term co-occurrence frequency for the $C_i$ (max – min). This will give us an integer value between an interval [0, 9]. We take a coarse granularity of 10 buckets, but this could be changed into more number of buckets if finer grained analysis is needed. Bucket algorithm gives us a set of importance of terms in a set of concepts for different search engines.

3.2 Algorithm based on Jaccard Coefficient

Another algorithm to calculate the relationship between $C_i$ and $t_j$ is based on Jaccard Coefficient [8].

Let $|C|$ denote the total number of web documents that include conceptual category $C_i$. Let $|T|$ denote the total number of web pages that include a term $t_j$. Let $|C_i \cup t_j|$ denote the number of documents where $C_i$ and $t_j$ co-occurs.

Degree of relationship strength between concept $C_i$ and $t_j$ is calculated by the following equation.

\[
W(C_i, t_j) = \frac{|c_i \cap t_j|}{|c_i| + |t_j| - |c_i \cap t_j|}
\] (2)

As an example, in a medical domain, let $SP_i$ be a concept $Specialty_i$ and $S_j$ be a term $Symptom_j$. In other words, $S_j$ is a term that describes a concept $SP_i$ in generic knowledge representation. By the Equation 2, we derive $W(SP_i, S_j)$.

\[
W(SP_i, S_j) = \frac{|no\_docs(SP_i \cap S_j)|}{|no\_docs(S_j)| + |no\_docs(SP_i)| - |no\_docs(SP_i \cap S_j)|}
\] (3)

Let’s assume we have 16 concepts (i.e., specialties) and 850 terms (i.e., symptoms) to be considered. Then,

\[
\begin{align*}
no\_docs(S_i) &= \sum_{t=1}^{16} |no\_docs(S_i)|, \\
no\_docs(SP_i) &= \sum_{t=1}^{850} |no\_docs(SP_i)|
\end{align*}
\]

3.3. Similarity Measure (Used by the Query Processor as the inference engine)

The similarity between a user query $q$ and concept $C_i$ is calculate by the Equation 4.

\[
\text{Similarity}(C_i, q) = \left[ \frac{C_i \cap q}{C_i \cup q} \right]
\] (4)

Let $n$ be the number of terms (descriptors) used to describe a concept $C_i$, and $a_i$ be the importance of each term $t_i$ assigned by the user, and $w_{ij}$ be the degree of association between term $t_i$ and concept $C_i$. Equation 4 can be rewritten as cosine similarity value between concept vector $C_i$ and query vector $q$, as shown in the Equation 5 [3, 4]. $Sim(C_i, q)$ in the equation 5 represents the similarity value of $C_i$ with respect to $q$.

\[
Sim(C_i, q) = \frac{\sum_{i=1}^{n} w_{ij} \cdot a_i}{\sqrt{\sum_{i=1}^{n} w_{ij}^2} \cdot \sqrt{\sum_{i=1}^{n} a_i^2}}
\] (5)

When we calculate the similarity value between concept $C_i$ and a user query $q$ by the Equation 5, we consider inverse concept frequency (ICF) for calculating $w_{ij}$ (the importance of term $t_j$ in concept $C_i$). ICF is based on the observation that the terms which rarely occur in the Knowledge Base (KB) are considered more informative.

Let $|C|$ be the total number of concepts (e.g., specialties) in the KB, and let $W(C_i, t_j)$ be the importance of term $t_j$ in the concept $C_i$ as we calculated before based on bucket algorithm and Jacaard Coefficient algorithm.

ICF based $w_{ij}$ is calculated by the following two equations:

\[
ICF = \log_{10} \frac{|C|}{W(C_i, t_j)}
\] (6)

\[
w_{ij} = W(C_i, t_j) \times ICF
\] (7)

4. Case Study: Nursing Home Application (NHA)
In developed countries, the aging population demand improvements on nursing homes, since as the concentration of elder people become prominent and people are busier than ever, better services need to be provided in order to maintain a good level of care to this aging population when they most need it.

In addition, since most of the nursing homes have limited information infra-structure, finding the most appropriate medical doctor with specialties who can take care of the patient’s emergency case is often difficult and not agile.

In this section, we present Nursing Home Application (NHA) as a reference implementation of the proposed generic architecture, on whose basis the knowledge based system (KBS) can be built based on web data analysis, as we explained in the previous sections.

NHA is a specialized application responsible for identifying which doctors (with specific specialties) are required based on a list of symptoms provided by the nursing home staff.

The idea is to quickly type in what symptoms the patient is feeling, and let the NHA decide which doctors should be notified based on that input. Once those are identified, the system broadcasts notifications to all available nursing home affiliated doctors for that specialty, trying to get an appointment as soon as possible. At that point, the system did its job and it’s up to the doctors and nursing home staff to arrange when and where the appointment will happen.

4.1 Nursing Home Application (NHA)

The NHA involves the development of two applications. First, a mobile client where nursing home staff can interact with the application, provide patients’ symptoms, and notify the emergency to the searched doctors. Second, a backend server that is responsible for providing web-services to be accessed by the client (mobile or desktop PC). The back end server implements the inference engine (refer to the section 3.3) that draws decision effectively and time efficiently by combining medical knowledge base system (KBS) automatically built (see sections 2 and 3). NHA will help nursing home staff members to identify the most appropriate doctors with specialties, and NHA automatically notifies the participating doctors to handle accordingly.

4.2 Building the Knowledge Base for the NHA

Our proposed architecture requires a quality knowledge base. As explained in sections 2 and 3, the most important data that is needed for building the knowledge base for NHA is the correlation between specialties (concepts) and symptoms (terms). The stronger a particular symptom is for a specialty, easier it is to identify the patient’s problem to converge into a specialty candidate and appropriate related doctors. At the same time, we consider inverse concept frequency to calculate
the relevant scores by the inference engine (see equation 6 and 7)

In order to construct a knowledge base we need to define what concepts and terms are important to the domain of study, where they could be acquired, and how. For the Nursing Home Application, we identified medical specialties and symptoms as the concepts and terms to be analyzed. Next, we researched on the web for websites that listed both of them, for convenience. Since we weren’t successful on trying to find both on the same document, we researched them independently and finally got two well-known resources: the American Board of Medical Specialties (ABMS), for our list of specialties, and Gemina, a project from the University of Maryland that studies symptoms and genomic sequences. Since the information was available on the website, after evaluating the HTML code of the web pages, we built scripts that parsed the content, saved into files that were then used as input to our knowledge acquisition system that is outlined in the Figure 2.1.

More specifically, first, we discovered a set of websites (Medical Institutions and Universities) that contained either a list of symptoms or a list of specialties. In order to extract data from these websites, we used JavaScript and the JQuery library to analyze the HTML structure, select the list of elements, and output the list to be exported/saved to a file. Using a simple standalone application, this file is then read and saved into the database for later use into building our knowledge base.

We start by checking the co-occurrence frequencies of both specialty and symptom from the well-known search engines. Co-occurrence frequency is the number of documents retrieved with respect to a pair of a specialty and a symptom. We could build knowledge bases based on collected symptom and specialty co-occurrence values based on various search engines. We collected such co-occurrence statistics for more than 1600 symptoms and more than 100 specialties.

For the NHA application, we built User Interfaces to interact with the mobile application, and web-based application.

### 4.3 Experiment Result for the Nursing Home Application

To evaluate the effectiveness of the knowledge base that was automatically acquired by the proposed architecture for the NHA application, nine queries are formulated as shown in Table 4.1. A query is formulated by extracting key words from the description of each disease from web sources (e.g., Alzheimer’s Disease from http://www.wakehealth.edu/Health-Encyclopedia/Health-Topics/Alzheimers-Disease.html)

<table>
<thead>
<tr>
<th>Disease</th>
<th>Relevant Specialties</th>
<th>Symptoms (Query)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brain Tumor</td>
<td>neurology or neurology with specialization Geriatric Medicine</td>
<td>Q1: headache, vomiting, inability to speak, loss of consciousness</td>
</tr>
<tr>
<td>Anemia</td>
<td>Hematology Neurology with Special Qualification Neurology Cardiology Diabetes and Metabolism</td>
<td>Q2: weakness, breathing problems, dizziness, fluctuation of heart rate, abnormal heart rhythms, headache, change in skin color, chest pain</td>
</tr>
<tr>
<td>Alzheimer's</td>
<td>geriatric medicine</td>
<td>Q3: memory loss, confusion, memory impairment, inability to form words, inability to think clearly</td>
</tr>
<tr>
<td>Osteoporosis</td>
<td>geriatric medicine</td>
<td>Q4: pain, joint pain, hip pain</td>
</tr>
<tr>
<td>IBS (Irritable Bowel Syndrome)</td>
<td>Gastroenterology</td>
<td>Q5: abdominal pain, gas pain, constipation, diarrhea</td>
</tr>
<tr>
<td>Pancreatitis</td>
<td>Gastroenterology</td>
<td>Q6: lesions in pancreas, loss of weight</td>
</tr>
<tr>
<td>Congestive Heart Failure</td>
<td>Cardiology</td>
<td>Q7: breathing problems, cough, wheezing, fatigue, leg swelling, abdominal swelling</td>
</tr>
<tr>
<td>Asbestosis</td>
<td>Pulmonary disease</td>
<td>Q8: breathing problems, chronic cough, chest pain, loss of appetite, abnormal chest sound</td>
</tr>
<tr>
<td>Asthma</td>
<td>Pulmonary disease</td>
<td>Q9: wheezing, breathing problems, cough, abnormal chest sound</td>
</tr>
</tbody>
</table>
We used three search engines to collect specialty symptom co-occurrence data for building knowledge base from the cyber space: Bing, AltaVista, and Blekko. For the performance evaluation of the effectiveness of the knowledgebase, we select 16 specialties out of 123 specialties for a simple illustration of the utility of the NHA application. The Table 4.2 is Specialty table each specialty is identified with identification number (ID). For each query, relevant specialties are found in the selected set of 16 specialties.

Table 4.2: List of Selected Specialties

<table>
<thead>
<tr>
<th>ID</th>
<th>SPECIALTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Cardiology</td>
</tr>
<tr>
<td>11</td>
<td>Cardiovascular Disease</td>
</tr>
<tr>
<td>12</td>
<td>Diabetes and Metabolism</td>
</tr>
<tr>
<td>13</td>
<td>Family Medicine</td>
</tr>
<tr>
<td>14</td>
<td>Gastroenterology</td>
</tr>
<tr>
<td>15</td>
<td>Geriatric Medicine</td>
</tr>
<tr>
<td>16</td>
<td>Geriatric Psychiatry</td>
</tr>
<tr>
<td>17</td>
<td>Gynecologic Oncology</td>
</tr>
<tr>
<td>18</td>
<td>Hematology</td>
</tr>
<tr>
<td>19</td>
<td>Internal Medicine</td>
</tr>
<tr>
<td>20</td>
<td>Internal Medicine - Critical Care Medicine</td>
</tr>
<tr>
<td>21</td>
<td>Neurology</td>
</tr>
<tr>
<td>22</td>
<td>Neurology with Special Qualification</td>
</tr>
</tbody>
</table>

The average recall and precision measures [7] are used to evaluate the effectiveness of knowledge base that were built by big data analysis based on our generic architecture. Precision is the ratio of the number of relevant specialties retrieved to the total number of irrelevent and relevant specialties retrieved among the selected 16 specialties. Recall is the ratio of the number of relevant specialties retrieved to the total number of relevant specialties in the knowledge base.

The relevant specialties with respect to the disease as a query Q1, as shown in the Table 4.1. The Table 4.2 shows the list of selected specialties to be considered for the experiment.

We calculate the average recall and precision values based on knowledge bases formulated by two different knowledge acquisition algorithms (Bucket algorithm, Jacaard Coefficient-based algorithm; see section 3). When similarity is calculated we used the Equation 5 and term co-occurrence and Jacaard tie-strength with ICF (Inverse Conceptual Frequency) as we explained in the section 3.3.

The following Table 4.3 shows the recall and precision values for the knowledge bases built based on analyzing big data collected from four search engines with respect to nine queries listed in the Table 4.2.

Table 4.3: Retrieval Performance in terms of Average Recall (AR) and Average Precision (AP)

<table>
<thead>
<tr>
<th>Search Engine and Algorithm</th>
<th>Q1 (AR, AP)</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
<th>Q6</th>
<th>Q7</th>
<th>Q8</th>
<th>Q9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bing (Bucket)</td>
<td>(0.67, 0.37)</td>
<td>(0.60, 0.29)</td>
<td>(0.63, 0.53)</td>
<td>(0.67, 0.66)</td>
<td>(1.00, 0.14)</td>
<td>(0.75, 0.16)</td>
<td>(0.63, 0.10)</td>
<td>(0.75, 0.42)</td>
<td>(0.75, 0.24)</td>
</tr>
<tr>
<td>Bing (Jacaard)</td>
<td>(0.67, 0.63)</td>
<td>(0.60, 0.58)</td>
<td>(0.63, 0.49)</td>
<td>(0.67, 0.58)</td>
<td>(1.0, 0.2)</td>
<td>(0.75, 0.27)</td>
<td>(0.63, 0.50)</td>
<td>(0.75, 1.0)</td>
<td>(0.75, 0.75)</td>
</tr>
<tr>
<td>Altavista (Bucket)</td>
<td>(0.67, 0.33)</td>
<td>(0.6, 0.48)</td>
<td>(0.63, 0.54)</td>
<td>(0.67, 0.32)</td>
<td>(1.0, 0.2)</td>
<td>(0.75, 0.13)</td>
<td>(0.63, 0.10)</td>
<td>(0.75, 0.23)</td>
<td>(0.75, 0.26)</td>
</tr>
<tr>
<td>Altavista (Jacaard)</td>
<td>(0.67, 0.52)</td>
<td>(0.6, 0.33)</td>
<td>(0.63, 0.33)</td>
<td>(0.67, 0.34)</td>
<td>(1.0, 0.5)</td>
<td>(0.75, 0.67)</td>
<td>(0.63, 0.58)</td>
<td>(0.75, 0.38)</td>
<td>(0.75, 1.0)</td>
</tr>
<tr>
<td>Blekko (Bucket)</td>
<td>(0.67, 0.37)</td>
<td>(0.6, 0.58)</td>
<td>(0.63, 0.38)</td>
<td>(0.67, 0.37)</td>
<td>(1.0, 0.09)</td>
<td>(0.75, 0.11)</td>
<td>(0.63, 0.82)</td>
<td>(0.75, 1.0)</td>
<td>(0.75, 0.33)</td>
</tr>
<tr>
<td>Blekko (Jacaard)</td>
<td>(0.67, 0.35)</td>
<td>(0.6, 0.61)</td>
<td>(0.63, 0.38)</td>
<td>(0.67, 0.53)</td>
<td>(1.0, 0.2)</td>
<td>(0.75, 0.75)</td>
<td>(0.63, 0.21)</td>
<td>(0.75, 0.33)</td>
<td>(0.75, 0.33)</td>
</tr>
</tbody>
</table>

The average recall and precision measures [7] are used to evaluate the effectiveness of knowledge base that were built by big data analysis based on our generic architecture. Precision is the ratio of the number of relevant specialties retrieved to the total number of irrelevant and relevant specialties retrieved among the selected 16 specialties. Recall is the ratio of the number of relevant specialties retrieved to the total number of relevant specialties in the knowledge base.

The relevant specialties with respect to the disease as a query Q1, as shown in the Table 4.1. The Table 4.2 shows the list of selected specialties to be considered for the experiment.

We could get effective retrieval performance with the automatically generated knowledge bases created based big data analysis based on our proposed knowledge acquisition architecture shown in the Figure 2.1. In general, Jacaard algorithm slightly outperforms the bucket algorithm for formulating the knowledge bases.

Bing based search performance is the best among the three search engines, which indicates more number of data
collected is more useful for building accurate knowledge base.

We could not use google search engine because of restrictions of using software generated queries. But with proper loyalty payment, we may be able to get more accurate knowledge base from google big document databases.

5. Conclusion

In this paper, we present generic architecture for building domain specific knowledge bases that can be automatically acquired by analyzing big data collected from the web environment. A list of concepts and terms (attributes) that describe concepts are assumed to be given or extracted from a set of web sites in a specific domain. We presented two different algorithms for analyzing big data for building knowledge base from the collected data sets. An algorithm for implementing query processor as an inference engine is also presented.

As a reference implementation of the proposed generic knowledge based search architecture, Nursing Home Application (NHA) was developed. Limited but meaningful experiment result is shown for validating the utility of the proposed architecture based on NHA as a case study.

Among the limitations of the NHA application is the dependency on user perception about their symptoms in order to take decisions, which might not exactly represent the reality, since users may have different perceptions and may omit important symptoms. In addition, the scale used to represent what the user is feeling may vary between patients, so even though they might be feeling the same thing, one might say their symptom seems to be very bad, while another might say it’s mild.

Obtaining quality big data is critical for building useful knowledge base for a domain specific application such as the NHA. Future work can be done to better analyze and enrich the knowledge base based on the semantic web analytics available in the Cyber space.

6. References


