Semantic Similarity Through Primitive Heterogeneous Query Classification in Medical Emergency and Remedy Engine

Prepared By:

Abhishek Saxena

12MCEC23

Internal Guide

Prof. Tarjni Vyas

Nirma University

External Guide

Mr. Satish Vagadia

J-KRI TechLabs, Pune.

Department Of Computer Science And Engineering

Institute Of Technology

Nirma University

Ahmedabad

May-2014

Semantic Similarity Through Primitive Heterogeneous Query Classification in Medical Emergency and Remedy Engine

Major Project

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer Science and Engineering

PREPARED BY:

Abhishek Saxena

12MCEC23

Internal Guide

Prof. Tarjni Vyas

Nirma University

External Guide

Mr. Satish Vagadia

J-KRI TechLabs, Pune.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD

CERTIFICATE

This is to certify that the Major Project entitled **Semantic Similarity Through Primitive Heterogeneous Query Classification in Medical Emergency and Remedy Engine** submitted by **Abhishek Saxena(12MCEC23)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science Engineering of Nirma University of Science and Technology, Ahmedabad is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, have not been submitted to any other university or institution for award of any degree or diploma.

Mr. Satish Vagadia
External Guide,
J-KRI TechLabs, Pune.

Prof. Tarjni Vyas
Internal Guide,
Institute of Technology,
Nirma University, Ahmedabad

Prof. Vijay Ukani
PG Coordinator - CSE,
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
HOD - CSE,
Institute of Technology,
Nirma University, Ahmedabad

Dr Ketan Kotecha
Director,
Institute of Technology,
Nirma University, Ahmedabad

Undertaking for Originality of the Work

I, **Abhishek Saxena**, Roll. No. **12MCEC23**, give undertaking that the Major Project entitled "**Semantic Similarity Through Primitive Heterogeneous Query Classification in Medical Emergency and Remedy Engine**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering** of Nirma University, Ahmedabad, is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student

Date:

Place:

DECLARATION

This is to certify that,

I, **Abhishek Saxena**, **12MCEC23**, a student of semester IV Master of Technology in Computer Science Engineering, Nirma University, Ahmedabad, hereby declare that the project work **Semantic Similarity Through Primitive Heterogeneous Query Classification in Medical Emergency and Remedy Engine** has been carried out by me under the guidance of Mr. Satish Vagadia, J-KRI TechLabs, Pune and Prof. Tarjni Vyas, Department of Computer Science and Engineering, Nirma University, Ahmedabad. This Project has been submitted in the partial fulfillment of the requirements for the award of degree Master of Technology (M.Tech.) in Computer Science and Engineering, Nirma University, Ahmedabad during the year 2013 - 2014.

I have not submitted this work in full or part to any other University or Institution for the award of any other degree.

Abhishek Saxena(12MCEC23)

ACKNOWLEDGEMENT

I would like to thank my Mentor, **Mr. Satish Vagadia**, J-KRI TechLabs, Pune for his valuable guidance. Throughout the training, he has given me much valuable advice on project work. Without him, this project work would never have been completed.

I would also like to thank my Internal guide **Prof. Tarjni Vyas**, Institute of Technology, Nirma University, Ahmedabad for her valuable guidance and continual encouragement throughout this work. The appreciation and continual support she has imparted has been a great motivation to me in reaching a higher goal. Her guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

I would also like to thank **Dr K Kotecha**, Director, Institute of Technology, Nirma University, Ahmedabad for providing me an opportunity to get an internship at J-KRI TechLabs, Pune.

I would like to thank my all faculty members for providing encouragement, exchanging knowledge during my post-graduate program.

The blessings of God and family members make the way for completion of Project. I am very much grateful to them.

Abhishek Saxena(12MCEC23)

Abstract

In medical emergency situations, providing the remedies and treatment in real time is important for saving someone's life. The target of this research work is to define a methodology, by which a person in need can be benefited. The research work is about processing the queries which are medically related and providing the remedies, immediate treatments and other useful information through which a person's life can be saved. The research project definition focuses on defining the methods and algorithms through which useful information can be retrieve from heterogeneous knowledge database and producing life saving information in real time. The query which will be passed to the engine will be native query. Algorithms will then be applied to this native query to further process the query, and make the decision which query need to be send to which database. Semantic similarities are then applied on the query and generate an output query, which then can be passed to heterogeneous knowledge databases to retrieve the useful data. The data retrieved would be similar to as a medical practitioner uses his judgments to get the remedies to the person. On the broader level we will have contact database, remedy database, current situation database. After receiving the data from different databases the data is processed and structured as defined in medical science, so that the person can take maximum benefit from the information generated. All these task and processing of the query should be done in real time. Present work focuses on optimization of the query and generating the result but no time constraint on the basis of the application domain.

This research work offers study of existing methods and algorithms and overcome their challenges of processing the queries and generate the algorithms which would generate the output in real time.

Contents

List of Figures

Chapter 1

Introduction

1.1 General

Medical emergency is a situation in which a person has a risk of its life being lost or an injury which can causes along term health problem. These emergencies or situation need to be handled immediately by the person himself or by some other person nearby .[1] The immediate remedy in such situation can be tieing up the bleeding area to stop bleeding, making sure that fellow colleague is in proper position to breath. Depending on the severity of emergency and the quality of immediate treatments given to the person, the person may further require the multiple levels of care, that is from normal physicians care to major operations.

In such situations instant remedy and treatment is must for saving a person's life. If a person is able to get medical practitioner like remedy at the time of emergency, then such information will be life saving for him as well for others. Defining the methodology and algorithms for providing such information in real time is main focus of this research work.

Just for example for better understanding the situations where the research work can be used, a person has met with an accident on highway and had stared bleeding heavily. In such situations instant remedy should be provided to that person for saving his life. Our research work focuses such situations. The person will raise the query for example, "met accident heavy bleeding". The query can be raise through SMS, API or through web. The location of the person will be identified first and then the query will be processed by the algorithms and will get the information like hospitals near by the

location, blood banks and other small clinics. The algorithm will also retrieve the immediate remedy from the database on the basis of the query and will also suggest some other symptoms for better understanding the situation, so that more specific remedy can be given to the person. Such result can be compared to as if a medical practitioner is providing the information and will be life saving for the person.By such information we are trying to increase the golden hour of the person.

1.2 Motivation

Motivation behind this research project is to help the people by providing the immediate treatments and remedy in emergency situations. There are emergency situations where every minute counts. By this research work we are trying to increase the "Golden Hour" of the person.

1.2.1 Golden Hour

Golden Hour or Golden Time, is a period of time lasting from a few minutes to several hours following an injury sustained by a casualty, and in this period it is strongly believed that immediate treatment will prevent the death of the person.[1]

This research work will also help the person by providing medical practitioner like treatments and remedy at any time. Research works aim in providing these remedies in real time.

1.3 Scope of the Work

To achieve the objective of proving the useful and life saving information to the user in real time, the work seek proper structuring of the query passed by the user and proper hierarchy of the databases wherein deciding which data need to be stored in which database. Defining the layers with in the databases and passing the right query to right database is the requirement of the research work.

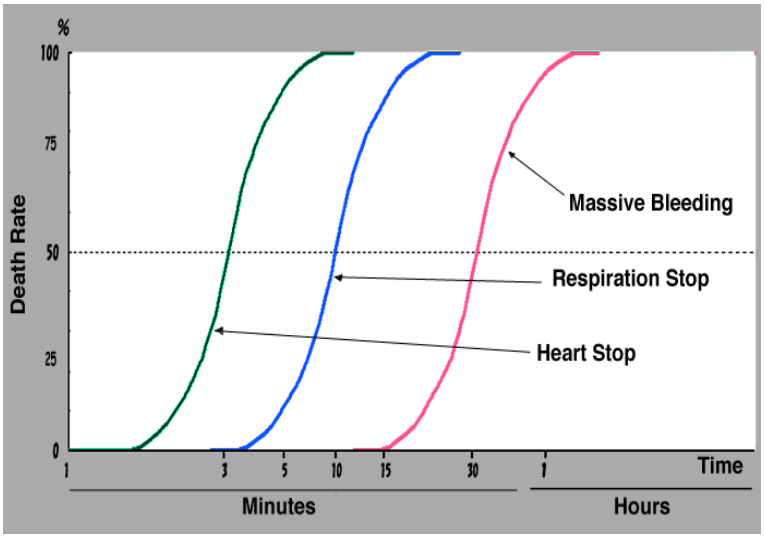

Figure 1.1: Time vs Death Rate
[1]

The scope of the project include proposing the database hierarchy and structuring the query so that fast and useful information can be retrieve from the database in real time.

Chapter 2

Medical Domain of Project

The scope of the project is restricted to medical domain particularly to medical emergency situations.

Medical emergency is a situation in which a person has a risk of its life being lost or an injury which can causes along term health problem. These emergencies or situation need to be handled immediately by the person himself or by some other person nearby .[1] The immediate remedy in such situation can be tieing up the bleeding area to stop bleeding, making sure that fellow colleague is in proper position to breath. Depending on the severity of emergency and the quality of immediate treatments given to the person, the person may further require the multiple levels of care, that is from normal physicians care to major operations.

2.1 Emergency situations

Medical emergency can occur at any time and at any place. Providing the assistant and right treatment is must to handle such situations.With the help of this research work a person can get such useful information in real time by raising the the query about what emergency need to be handled.

For example if a person met with an accident and started bleeding heavily, in such situation immediate assistant and initial treatment is must. Project will help in such situation by providing the immediate treatments which will help to stop bleeding, information of

hospitals near by and information about the blood banks.

Not only in the emergency situation, the project will be very useful, if a person just need to know what are the first measure to handle a simple fever.

Some warning signs of a medical emergency are :[2]

Choking	Chest pain
Bleeding that will not stop	vomiting blood or blood while coughing
Loss of consciousness	Head or spine injury
Persistent vomiting	Change in mental status
Intake of poisonous substances	Upper abdominal pain

Chapter 3

Information Retrieval Models

3.1 Boolean Model of Information Retrieval

The Boolean model of information retrieval (BIR)[6] is an information retrieval model which is based on boolean logics and set theory, In this model both the query given by the user and the documents in which searching is to be performed are taken as a set of terms. The retrieval of the data is based on whether the documents in which searching is performed contain the query term or not.

Given a finite set of index term,

T = t1, t2, t3, ..., tn

Also finite set of documents which is power set of terms T,

D = D1, D2, D3, ..., Dm,

Boolean expression of query Q is as follows,

Q = (Zj OR Zi OR Zk ...) AND (Zp OR Zq OR Zr OR ...), with Zi=ti, Zk=tk, Zj=tj, Zq=tq, or Zi=non ti, Zk=non tk, Zj=non tj, Zq=non tq

where tq means the term tq is present in the document Dq, whereas non tq means that it is not.

Retrieval operation can be performed in two steps:

1. The Document sets Sr is obtained that contain the term tr (whether Zq=tq or Zq=NON tq) :

Sr = Dq—Zq element of Dq

2. The documents which are answere to Q are retrieved as follows:
   UNION (INTERSECTION Sr)

3.1.1 Advantages

1. Simple and Clean formalism of algorithm.

2. Easy to implement.

3.1.2 Disadvantages

1. Unnecessary documents are retrieved from exact matching.

2. Ranking of the output is difficult. Raking is important as it filters out the unimportant documents.

3. Transformation of query to Boolean expression is hard.

4. It is more like data retrieval than information retrieval.

3.2 Vector Space Model

Vector space model is model in which the text documents are represented as vectors of identifiers.It is an algebraic model. It is an important model for information retrieval, filtering and for document ranking.

We represent the documents and queries as vectors as follows:[11].

$$D_i = (W_{1,i}, W_{2,i}, W_{3,i} \ldots, W_{n,i})$$
$$Q = (W_{1,r}, W_{2,r}, W_{3,r} \ldots, W_{m,r})$$

The dimensions of the vector correspond to a different term. The value of the vector is non zero if the terms occurs in the document .We can compute these values in different ways but on of the popular and best known scheme is tf-idf weighting.
We can define term in various ways and it majorally depends on the application. A term can be longer phrases, keywords or a single word. Depending on the terms the dimension of the vector changes.

3.2.1 tf-idf Weighting

term frequencyinverse document frequency (tfidf), is a statistical number which indicate the importance of a word in a document[10].
It is an important weighting factor in text mining and information retrieval. As the number of times word appearers in the document increases the tf-idf value also increases propotrtionally, but it also takes care about the common words which appears more in the documents

tfidf is define as the product of two statistics, term frequency and inverse document frequency[10]. The term frequency tf(T,D), is define as the number of times a word appears in the document i.e. simply the word count.

The inverse document frequency is a measure of the rareness of the word in the documents i.e.how common or rarer is the word in all documents. Mathematically it is defined as the logarithmic scaled fraction of documents that contain the word. It is obtained by taking the fraction of total number of documents to the number of documents containing the term, and then taking the log of it.

$$idf(T, D) = \log \frac{|d|}{|\{D' \in d | T \in D'\}|} \tag{3.1}$$

with $\log \frac{|d|}{|\{D' \in d | T \in D'\}|}$ is inverse document frequency which is a global parameter.
$|d|$ is the total number of documents in the document set.
$|\{D^{'} \in d | T \in D^{'}\}|$ is the number of documents containing the term T.

Then tfidf is calculated as

$$W_{T,D} = tf(T, D) \times idf(T, d) = tf_{T,D}. \log \frac{|d|}{|\{D' \in d | T \in D'\}|} \tag{3.2}$$

3.2.2 Advantages

1. The model is based on linear algebra which is simple to use.
2. It allows to compute the similarity between queries and documents on continuous degree.
3. It helps in ranking the documents.

3.2.3 Disadvantages

1. Model lacks in representing the long documents properly majorally because of similarity value.
2. There is a problem of "false positive match"
3. Problem of "false negative match". Model won't associate the document with similar context but different term vocabulary.
4. The term order is lost in this model, i.e. the order in which the term appears in the document.

3.3 gGLOSS

The Glossary-of-Servers Server (GlOSS), as an approach by which the problem of database selection can be solved[4]. It is basically used to handel the boolean model. Later we have the generalized version of GLOSS i.e. gGlOSS which can handle both boolean model as well as vector space model. This approach helps us to find the goodness of the document with the query, that is how good is the document for satisfying the query.

gGlOSS approach makes an assumption that we can characterized the document according to their goodness with the query. gGlOSS's then create a priority list of the documents according to the goodness of the query. This ranking helps us to remove or ignore the documents which are not usefull to the query.
We can calculate the Goodness of each database, DB, as follows:

$$Goodness(l, Q, DB) = \sum_{d \in rank(l,Q,DB)} sim(Q, D)[4] \tag{3.3}$$

where sim(Q,D) gives the similarity meter between query Q and document D. $rank(l, Q, DB) = \{D \in DB | sim(Q, D) > l\}$
By sorting the database on the basis of their goodness we can find the ideal(l) rank of database for query Q.

We define an important function through which similarity between document D_i and query Q is calculated:

$$sim(D_i, Q) = \frac{\sum_{j=1}^{n} W_{j,i} W_{j,r}}{\sqrt{\sum_{j=1}^{n} W_{j,i}^2} \sqrt{\sum_{j=1}^{n} w_{j,r}^2}}[4] \tag{3.4}$$

Chapter 4

Literature Survey

Luis Gravano Hector Garcia-Molina, enlighten the problem of identifying the correct databases for evaluating the given query.[4]In the real world there are millions of documents and sources available and out of those documents finding the right one for our query is practically impossible.

In their paper they presented gGlOSS, an approach that will rank the databases according to their goodness with the query and will provide a priority list of documents to which we can fire our query.This statistical information is very helpful in reducing the search time.

John Grant, Jarek Gryz and Jack Minker [7], demonstrate the technique for optimization of the user query. This technique i.e. semantic query optimization (SQO) technique helps in optimizing the query by using the semantic knowledge, and reformulate a query. This semantically equivalent query, can be evaluated more efficiently and thus will save time in information retrieval.

They further propose that, two queries are semantically equivalent if they obtain the same answers in the database that satisfies the same set of integrity constraints.

Chun-Nan Hsu and Craig A. Knoblock [8], has explore the need for integrating large set of heterogeneous database for retrieval of data for health care information system over computer network.To answere the query in such situation require to find the relevant database and then method to combine the data.

Majority of work has been done in selection of source and generation of plan but

critical issue lies in query optimization, which will help in retriving the data and selecting the source efficiently. This paper presents SQO approach i.e. semantic query optimization approach which will help in optimizing the query for heterogeneous database environment.

This approach has two plans for query optimization, global optimization and local optimization. Global optimization plan is for query and local optimization is for subqueries which are used to retrieve the data from individual databases. The important feature of local optimization algorithm is that it eleminate the unnecessary joins in a query by proving sufficient conditions. This feature further enhance our optimizer and allow it use expressive relational to further optimize the query for efficient retrieval.

Jie Shen, Guisheng Yin, Xiaomei Ma [9], has put the light on how to process the query on mass data.The user can change their query and modify it according to the results which they have gained dynamically as an intermediate result of the query. This will help them to retrieve the data more accurately.

Rada Mihalcea and Courtney Corley and Carlo Strapparava,[13] presents an approach for calculating the semantic similarity of the texts. Today we have a large set of information available on web and these information are generally in the abstract form i.e. of short text. This papaer focus on finding the simantic similarity of these short text and rank the database according to the query.

Chapter 5

The Proposed Algorithm

Considering the scope of the project, the query raise by the user will be in natural language and restricted to medical domain.By taking the above facts in mind proper structuring and preprocessing of the query need to be done, so as to achieve fast and efficient retrieval of the data from the database.

5.1 Query Structuring

The part of my research work will receive the native query that has been converted from the query raise by the user in natural language.

For example:

Query Raise : **"fever from 3 days"**

Native Query :{

Location:{

'Place' : 'Pune,Maharashtra,India',

'Latitude' : '19.0094',

'Longitude' : '73.120'

},

'Time' : '3 PM',

'Date': '15-11-13',

'Gender' : 'NA',

'Age' : 'NA',

'Action' : 'Fever',

'Outcome' : '3 Days',

'Category' : 'Critical',

}

On the basis of this native query different queries will be generated to extract the information from heterogeneous database.

5.2 Sample Query Formation

The algorithm will take the native query as an input and will structure different queries. These query will then be fired to different different databases to retrieve the data. The algorithm works on the basis of weightage of the keys in the native query. For example "Category" key can have 3 weightages minor, major and critical. Minor is the lowest weightage, then comes major and critical has highest weightage, and accordingly the data will be stored in the database.

5.2.1 Query to Contact Database

According to the location array in native query, which contain the information of the place, a sperate query will be generated and will be fired to contact database.

Select Name, Address from contact database where longitude(in range from longitude -5 to longitude +5) and latitude (in range from latitude -5 to latitude +5)

5.2.2 Query to Remedy Database

According to the key "Category","Outcome" and "Action" in native query, a sperate query will be generated and will be fired to remedy database.

Select Description from Remedy Database where category= 'Category' and outcome='Outcome' and action='Action'

5.3 Ranking the Databases

The algorithm uses vector-space databases and queries.For query, we will rank the available vector-space databases depending on their usefulness. This ranking should give us the ideal order of the databases for searching the information.The algorithm maintain the statistics on the databases, and uses these statistics to estimate the actual contents of the databases.

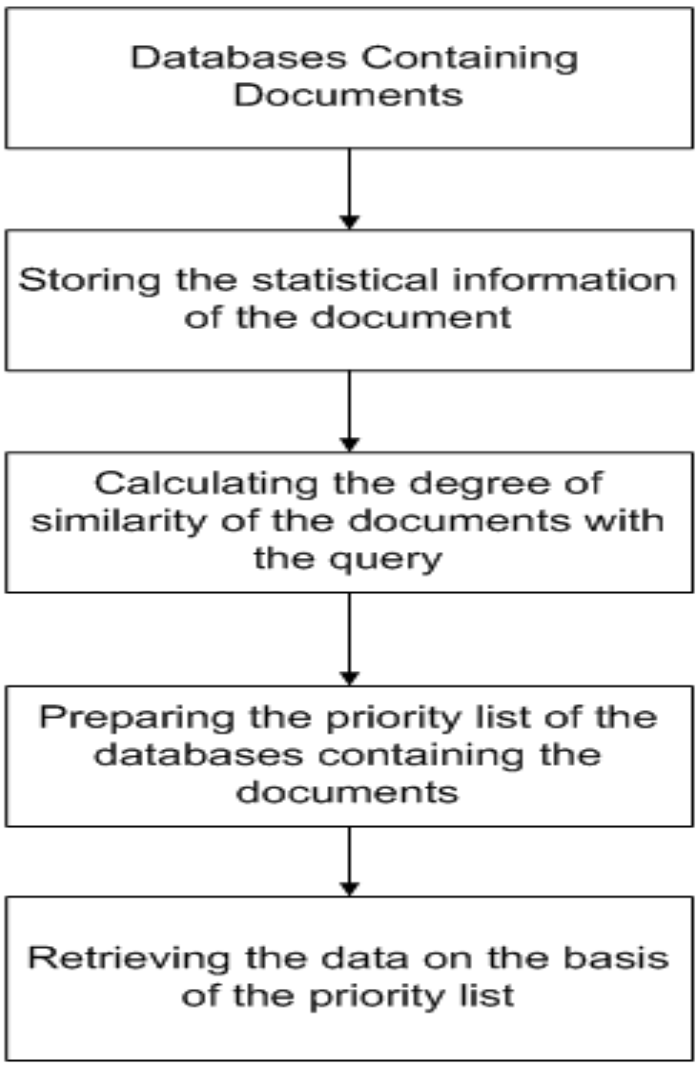

Figure 5.1: Algorithm flow

For each and every database we calculate a Goodness factor which determines how

good each document in database db is useful for query q. We consider only those documents in db useful for q whose similarity to query is greater than a given threshold l. Documents with similarity less than threshold are not considered useful and are ignored.

Goodness of the database can be calculated as

$$Goodness(l, Q, DB) = \sum_{D \in rank(l,Q,DB)} sim(Q, D)[4] \tag{5.1}$$

where sim(Q,D) gives the similarity meter between query Q and document D. $rank(l, Q, DB) = \{D \in DB | sim(Q, D) > l\}$
By sorting the database on the basis of their goodness we can find the ideal(l) rank of database for query Q.

We define an important function through which similarity between document D_i and query Q is calculated:

$$sim(D_i, Q) = \frac{\sum_{j=1}^{n} W_{j,i} W_{j,r}}{\sqrt{\sum_{j=1}^{n} W_{j,i}^2} \sqrt{\sum_{j=1}^{n} w_{j,r}^2}}[4] \tag{5.2}$$

5.3.1 Example

For instance consider two databases, DB_1 and DB_2 and a query Q. The answers that two databases give when presented with query Q are:

$DB_1 : (D_1^1, 0.8), (D_1^2, 0.8), (D_1^3, 0.2)$
$DB_2 : (D_2^1, 0.7), (D_2^2, 0.3), (D_2^3, 0.1), (D_2^4, 0.2)$

In above example, DB_1 returns the documents DB_1^1, DB_1^2 and DB_1^3 as a answer to query Q.

Documents DB_1^1 and DB_1^2 are ranked highest, as they are closest to the query Q in databases DB_1 with similarity 0.8.

Let the threshold be l = 0.1. According to goodness equation

$$Goodness(0.1, Q, DB_1) = 0.8 + 0.8 = 1.6$$

Similarly we have Goodness for database 2:

$$Goodness(0.1, Q, DB_2) = 0.7 + 0.3 + 0.2 = 1.2$$

This goodness metric helps in identifing how good/useful a database is for the user query. Therefore, Ideal(0.1) is DB_1, DB_2.

5.4 Defining Optimizing Parameters

Algorithm calculates the goodness information and stores it on the available databases, to further estimate the usefulness of the database with query. One way is to store the complete information on every database. To keep this full information we required a large storage on the databases.So to reduce the storage requirement one option is to keep incomplete but useful information only, on the databases.For this we require the following matrices:

$N = (n_{ij}) : n_{ij}$ [4]is the numerical value which tell the number of documents in database DB_i that contains the word w_j

$F = (f_{ij}) : f_{ij}$ is the numerical value which is the sum of weights of word w_j over all documents present in the database DB_i

On the basis of these parameters we make certain assumptions to optimize our algorithm

5.4.1 Assumption 1

If query keywords w1 and w2 appears in ni1 and ni2 documents in database DBi, respectively, and ni1 and ni2, then every DBi document that contains w1 also contains w2.[4]

For example consider a database DB_i and the query Q=accident heavy bleeding.

w1= accident, w2= heavy, and w3= bleeding.

$n_{i1} = 3$, $n_{i2} = 10$, and $n_{i3} = 11$: i.e. there are 3 documents in DB_i with the word "accident", 10 with word "heavy", and 11 with word "bleeding".

Algorithm assume that the 3 documents with word "accident" also have the words "heavy" and "bleeding". Furthermore, all the $10 - 3 = 7$ documents having the word "heavy" but not the word "accident" also contain the word "bleeding".Finally, there is exactly $11 - 10 = 1$ document with only word "bleeding".

5.4.2 Assumption 2

This assumption state that the weight of the word is uniformly distributed over all the documents containg that word. Word w_j has weight $\frac{f_{ij}}{n_{ij}}$ in every DB_i document that contain the wordw_j.

For instance continuing from previous example, Suppose the total weight of the query words in database DB_i are f_{i1}= 0.46, $f_{i2} = 0.3$, and $f_{i3} = 0.10$.

The three documents that contains the word "accident" will have the weight equals to$\frac{0.46}{3} = 0.153$. The ten documents that contains the word "heavy" will have the weight weight equals to $\frac{0.3}{10} = 0.030$.

5.5 Defining Max Threshold l

The above assumptions are used by the algorithm to estimate the number of documents in a database having similarity with the query Q, greater than the threshold l. Algorith also calculate the total similarity of these documents, to estimate the Max(l) database rank.

5.5.1 Example

For example assume the query Q has the weight 2 for each of its three words. According to Assumption 1, the three documents with word "accident" also contain the words "heavy" and "bleeding" in them. The similarity of these three documents to Q is, according to the assumption is$\frac{0.46}{3} + \frac{0.3}{10} + \frac{0.10}{11} = 0.192$.

If for example our threshold l be 0.1, then according to our algorithm we can accept all these documents as their similarity to Q is higher than 0.1. Further there are $10 - 3 = 7$ documents having word "heavy" and "bleeding" but not "accident".The similarities of any of these 7 documents to Q can be calculated as $\frac{0.3}{10} + \frac{0.10}{11} = 0.039$. Then according to our algorithm we cannot accept these documents for threshold $l = 0.1$. Also There are $11 - 10 = 1$ document containing only the word "bleeding", but this document cannot be accepted as its similarity to Q is even lower.

Finally with these metrics we have a list of priority databases from which we need to fetch the data.

Chapter 6

Experimental Result

Experiments are perform by taking first small number of documents and then gradually increasing the number of documents and Examining the different parameter.

Algorithm perform well for small databases but for large databases the goodness metric decreases slightly.

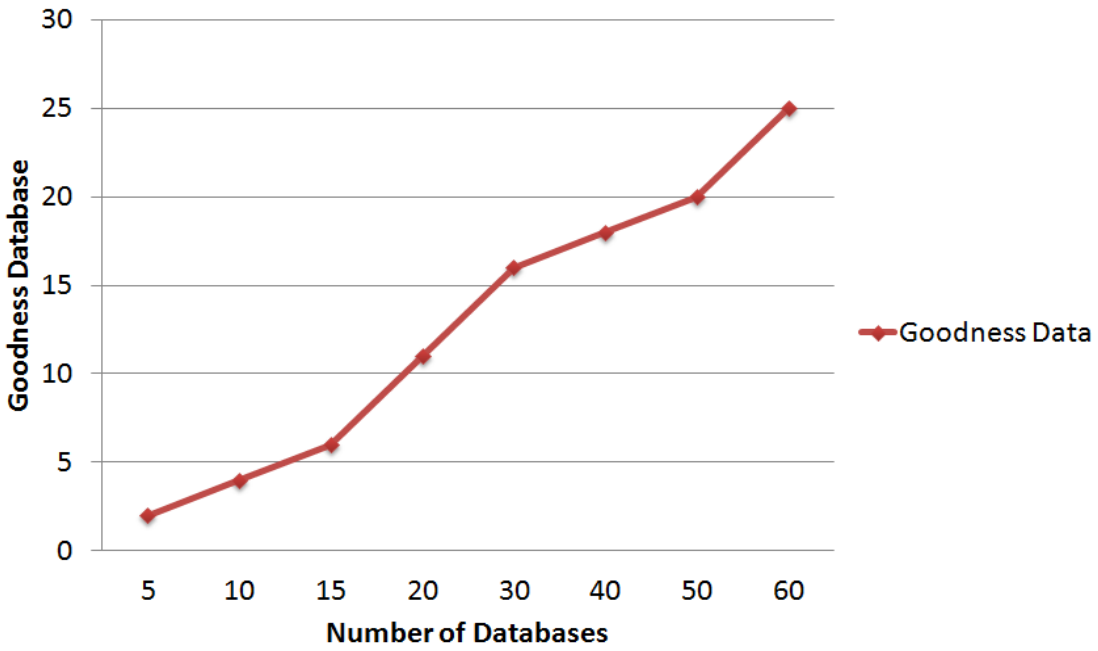

Figure 6.1: Graph of Ranked Database Retrieved

Graph showing the maximum similarity of the database with query.As from the graph it is clear that similarity increases as the number of databases increases

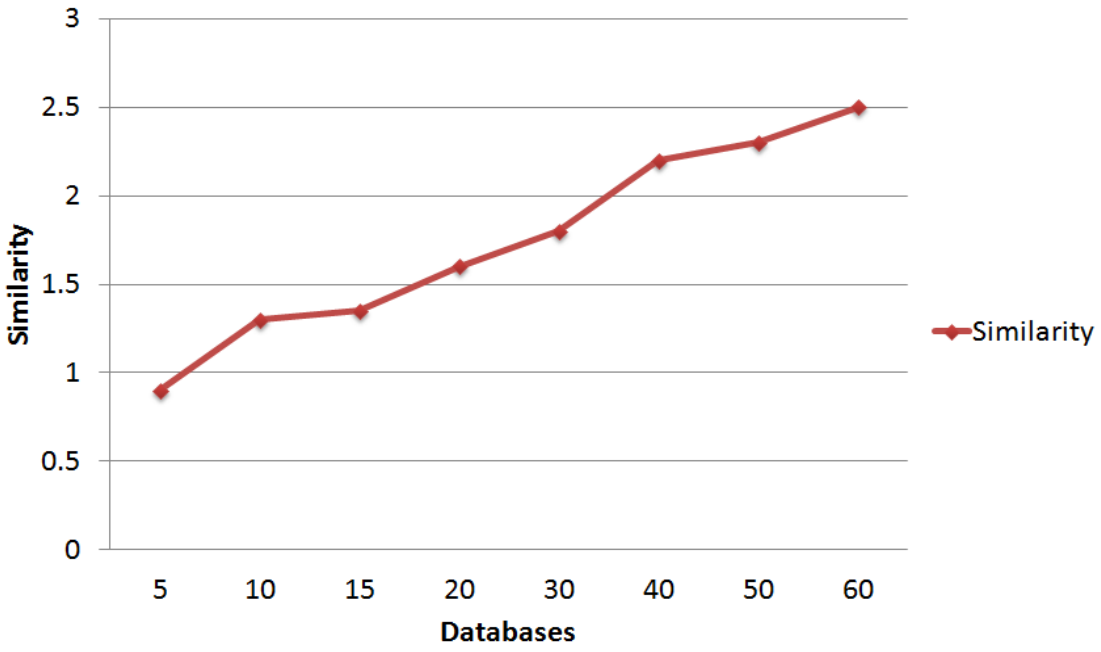

Figure 6.2: Graph of Similarity vs Database

Time increases as the number of priority document increases to be searched.

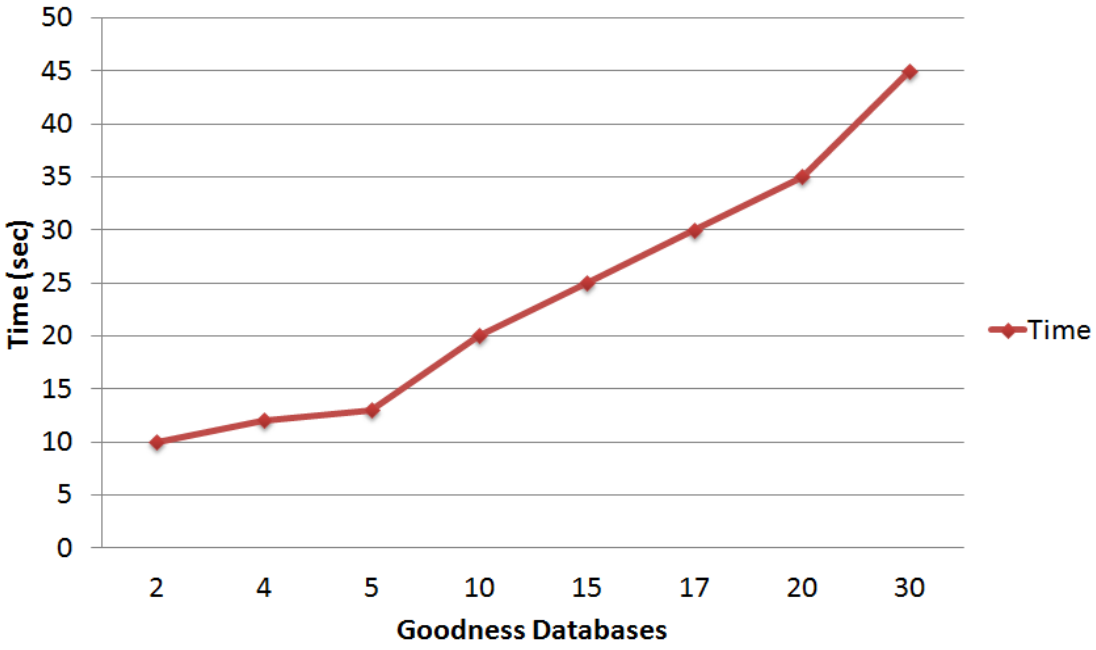

Figure 6.3: Graph of Time vs Database

Chapter 7

Conclusion and Future Work

This report propose an algorithm for ranking the database on the basis of the similarity of the database with the query fired.Through this algorithm we will get the priority list of the database through which we can retrieve the required information.This ranking helps to reduce the time required to retrieve the information from heterogenous databases and also only the useful information is retrieved.

Future Work

- Extending the algorithm for bigger databases.
- Still some optimizing of the algorithm is required to achieve more accurate and fast retrieval.

Bibliography

[1] http://www.medindia.net/patients/patientinfo/traumagoldenhour.htm

[2] http://www.nlm.nih.gov/medlineplus/ency/article/001927.htm

[3] http://technet.microsoft.com

[4] Generalizing GlOSS to Vector-Space Databases and Broker Hierarchies Luis Gravano Hector Garca-Molina Computer Science Department Stanford University Stanford,

[5] Bicsi, B., (2002). Network Design Basics for Cabling Professionals. City: McGraw-Hill Professional

[6] A Boolean Model in Information Retrieval for Search Engines,Lashkari, A.H. FCSIT, Univ. of Malaya (UM), Kuala Lumpur, Mahdavi, F.Ghomi

[7] Logic-Based Query Optimization for Object Databases John Grant, Jarek Gryz, Jack Minker, Fellow, IEEE, and Louiqa Raschid, Member, IEEE

[8] Semantic Query Optimization for Query Plans of Heterogeneous Multidatabase Systems, Chun-Nan Hsu and Craig A. Knoblock, IEEE Transaction on Knowledge and data engineering, Vol. 12, No. 6, November/December 2000

[9] Incremental Optimization Query in XPath with the Tree Automaton, Jie Shen, Guisheng Yin, Xiaomei Ma, 2009 International Symposium on Information Engineering and Electronic Commerce

[10] An improved TF-IDF weights function based on information theory ,Na Wang Dept. of Electron. Commun., Zhengzhou Inst. of Aeronaut. Ind. Manage., Zhengzhou, China Pengyuan Wang Baowei Zhang

[11] On N-layer Vector Space Model-Based Web Information Retrieval,Weiqun LuoSch. of Inf. Eng., Tibet Inst. for Nat., Xianyang, China, Chungui Liu, Zhiwei Liu, Conghua Wang

[12] An Algorithm for Answering Queries Eciently Using Views Prasenjit Mitra Infolab, Stanford University Stanford, CA, 94305, U.S.A. mitra@db.stanford.edu September, 1999

[13] Corpus-based and Knowledge-based Measures of Text Semantic Similarity Rada Mihalcea and Courtney Corley Department of Computer Science University of North Texas, Carlo Strapparava Istituto per la Ricerca Scientifica e Tecnologica ITC irst