ABSTRACT

Keyword search is a process of searching for relevant documents on the Web using one or more user specified words called Keywords. Keywords and their related data elements are linked using keyword elements relations. It is a method of querying linked data sources on the Web. These queries search for the related data over all relevant sources on the Web and present a lot of suggestions, of which many are unnecessary. We can reduce the number of results that are not relevant by keyword combinations in the query, but this makes it difficult to handle the query efficiently. It also increases the response time of the query which is not desirable in today's Web scenario demanding high responsiveness.

To reduce this high-cost of processing the query, a novel method is proposed to route the keywords only to relevant sources over all sources. Routing keyword search is a novel proposal to improve the performance of the keyword search and helps in minimizing the time and space costs.
# TABLE OF CONTENTS

Abstract........................................................................................................................................1

Table of Contents..........................................................................................................................2

List of Figures..................................................................................................................................4

1. BACKGROUND AND RATIONALE.........................................................................................6
   1.1 Introduction............................................................................................................................6
   1.2 Linked Data............................................................................................................................7
   1.3 Resource Description Framework (RDF)............................................................................9
   1.4 Data Mining...........................................................................................................................9
   1.5 Existing System.....................................................................................................................10
   1.6 Proposed System...................................................................................................................11

2. NARRATIVE...............................................................................................................................13
   2.1 Problem Statement...............................................................................................................13
   2.2 Motivation.............................................................................................................................13
   2.3 Project Scope.......................................................................................................................14

3. SYSTEM DESIGN AND ARCHITECTURE...............................................................................15
   3.1 Architecture.........................................................................................................................15
   3.2 System Requirements..........................................................................................................21
      3.2.1 Functional Requirements..............................................................................................21
      3.2.2 Software Requirements...............................................................................................22
      3.2.3 Hardware Requirements..............................................................................................22

4. SYSTEM IMPLEMENTATION.................................................................................................23
   4.1 Environment..........................................................................................................................23
4.1.1 Java Server Pages (JSP) ................................................................. 23
4.1.2 Java Development Kit (JDK) ....................................................... 23
4.1.3 Java Springs .............................................................................. 24
4.1.4 MyEclipse .................................................................................. 24
4.2 Web of Data .................................................................................. 24
4.3 User Interface ................................................................................ 25
  4.3.1 Existing Search ......................................................................... 25
  4.3.2 Proposed Search ........................................................................ 28
  4.3.3 Enhanced Search ....................................................................... 31
5. TESTING AND EVALUATION ............................................................ 33
  5.1 Unit Testing .................................................................................. 33
  5.2 System Testing ........................................................................... 33
  5.3 Test Cases ................................................................................... 34
6. CONCLUSION .................................................................................. 44
7. Bibliography ................................................................................... 45
LIST OF FIGURES

Fig. 1.1: Part of the Linking Open (LOD) data project cloud diagram 7
Fig. 3.1 Architecture of the proposed system 18
Fig. 4.1: Fig. 4.1: Searching for the keyword “edison” in existing search 26
Fig 4.2 Search results for the keyword Edison in existing search 26
Fig: 4.3 Triples generated for the keyword “Thomas Edison” 27
Fig: 4.4 Triple sequence generated for keyword “Thomas Edison” 23
Fig.4.5 Triples generated for keyword “Thomas Edison” proposed search 29
Fig 4.6 Final Triples sequences generated in proposed search 31
Fig 4.7 Querying the enhanced search 32
Fig 4.8 Results generated with count values 32
Fig 4.9 Results after applying a threshold on the results 33
Fig 5.1: Search results for the keyword “Sachin” 36
Fig 5.2: Triples generated for the keyword “Sachin” 37
Fig 5.3: Final triple sequence for the keyword “Sachin” 38
Fig: 5.4 Triples generated in the proposed search for the
keyword “Sachin” 39
Fig: 5.5 Triples sequence generated in proposed search for
keyword “Sachin” 39
Fig: 5.6 Final triple sequence generated in the proposed search for the
keyword “Sachin” 40
Fig. 5.7 Querying the enhanced search 41
Fig. 5.8  Search results with estimation values in enhanced search  41
Fig. 5.9  Search results with count values in enhanced search  32
1. BACKGROUND AND RATIONALE

1.1 Introduction

The Web today is not only a collection of textual data but also a collection of interlinked data sources (e.g., Linked Data). Linking Open Data is one such large project through which large amount of legacy data is transformed into the Resource Description Framework (RDF) and linked to other sources and published as linked data [1]. Linked data is comprised of many sources that contain billions of Resource Description Framework triples which are linked by millions of links like 'sameAs' links, which are published more frequently.

It would be difficult for a typical web-user to explore this linked data on the Web using any structured query languages. This is where the keyword search is applied. Unlike structured query languages, here, it is not necessary for the user to have any knowledge of the schema of the underlying data that he need to exploit. In the present scenario when a query is passed to the database through a keyword, it searches for the most relevant structured results [1], [2], [3] or a single relevant database. The issue with this approach is the Web of Linked Data is not directly applicable as a source may encompass may Linked sources of data. The main problem with this approach is not about finding the most relevant source, but computing most relevant combination of sources[6],[7]. We propose to generate a routing plan that can compute the results from multiple data sources.

1.2 Linked Data

Linked Data provides a description for the method of publishing structured data for the purpose of interlinking and making the structured data more useful through semantic queries.
Related documents and related data are linked on the Web. Linked Data defines the set of best practices for connecting structured data and publishing it on the Web [15]. Linked data is built on standard Web technologies like HTTP, URI and RDF [14]. Rather than using these technologies to just serve web pages for user requests, linked data employs them to share information in such a way that computers can directly read it. Thus data from different sources is connected and can be queried. Linked data describes how the Web is used for connecting related data that was not previously connected and lowering the barriers of linking the data that is linked currently by using other methods [15]. Fig 1.1 shows an overview of how the data from different datasets is connected in the Web.

Fig 1.1: Part of the Linking Open (LOD) Data Project Cloud Diagram [21]
1.3 Resource Description Framework (RDF)

The Resource Description Framework (RDF) is a set of specifications designed by the World Wide Web Consortium (W3C) as a metadata model [16]. The Resource Description Framework is generally used in the Web resources for conceptual description and modeling of information. It is similar to traditional approaches used for conceptual modelling like class-diagrams or entity relationships but is mainly used to describe relation between the Web resources. In Resource Description Framework the relations are expressed as triples in the form of subject–predicate–object. Here, the subject denotes a resource, the object denotes the information of the subject and the predicate describes the relation between the subject and the object. In a simple way, we can say that a predicate is an edge between the two nodes, the subject and the object. The subject and objects can be swapped like in the classical notation of entity-attribute-value model in object oriented design where object is object, subject is attribute and predicate is value. Collection of Resource Description Framework can be represented as a labeled directed multi-graph [16]. Hence a data model based on Resource Description Framework is more suitable for certain knowledge representations than tradition entity-relation model or other ontological models.

1.4 Data Mining

Though the term data mining was coined in 1990's, the concept of data mining dates back many years. The growth of Data mining began with the beginning of data storage on computers. Data mining evolved with the advancements in computer technology like data storage, processing power of the computers, new software’s and new algorithms. However the major advancements in
data mining happened with the introduction of relational databases and structured query languages. The next improvement came with the evolving of data warehousing and online analytic processing.

Data Mining is the process of knowledge extraction from large sets of data by analyzing the data and discovering consistent patterns and semantic relations between variables [13]. This knowledge is used to validate the findings by applying the detected patterns on new data. The study of Data mining involves artificial intelligence, statistics, machine learning and databases. Data mining mainly has three phases Exploration, Pattern Identification and Deployment.

The exploration deals with preparing of data that involves cleaning the data, transforming it and selecting subsets of records from the data by performing some preliminary operations based on the requirements. The second phase also known as model building, considers various models and choose the best one based on their predictive performance. A variety of techniques are developed to attain this goal based on competitive evaluation of models. Some of these techniques are Bagging, Boosting, Stacking and Meta-Learning [13]. The last phase, the deployment phase, utilizes the model chose as the best in the previous phase and applies it to the new dataset to produce predictions and estimations.

1.5 Existing Systems

The current study on keyword querying is in two different directions. The first study mainly focuses on the search approach computing the most relevant structured results and the later study focuses on source selection to compute the relevant source [1].
Many number of frameworks have been designed previously to produce keyword query results. These frameworks, when given a keyword query, retrieve the most relevant structured results, or simply, select the single most relevant databases. However, these approaches are single-source solutions. They are not directly applicable to the Web of linked data, where the results are not bounded by a single source but might encompass several linked data sources. As opposed to the source selection problem, which is focusing on computing the most relevant sources, the problem here is to compute the relevant combination of sources.

When a keyword is queried in the existing system, it searches the relevant results and generates routing plans for the obtained results and displays them all. The quantity of potential results may increment exponentially with the number of sources and the links connecting them. Most of the results for such queries may be redundant, particularly when the query is simple and the resulting links connected to that keyword are more. The routing problem, we need to compute results capturing specific elements at the data level. Routing keywords return the entire source which may or may not be the relevant sources.

**Disadvantages:**

The following are the major drawbacks of the existing approach which can be minimised by implementing minor changes in the existing approach.

1. With the increase in the number of sources and links connecting them the potential results may also increase exponentially and most of the results may not be useful when they are not relevant to the user query.
2. Computing results to capture elements at data level is the actual routing problem.

3. Routing keywords usually return entire source that may or may not be a relevant one.

1.6 Proposed System

We propose a new method to solve the problem of keyword search over a large number of linked and structured data sources using keyword query routing. The high-cost of searching for keywords that span across different sources can be reduced by routing the keywords only to relevant sources. Unlike the existing system which only uses the relationships between the keywords, we employ the keyword element relationship graph [9] and apply routing plans over the obtained results. Then we apply Maximum Likelihood algorithm on the obtained results to minimize the number of results by filtering the unwanted results we obtained from the keyword element relationship graph.

**Advantages:**

The following are the advantages of the proposed system.

1. Possible to reduce the cost of the search.

2. Possible to reduce the time for the search.

3. Produce the results from multiple resources.
2. NARRATIVE

2.1 Problem Statement

Semantic data query expansion procedure is important in the information retrieval process. Now here we focus on query optimization for semantic data extraction. Neighbor likelihood algorithm or Maximum Likelihood algorithm discovers the semantic keywords to generate the efficient query. Using semantic keywords we construct the enhanced keyword query with good relationships. We call enhanced keyword query pattern as an optimal query plan as it produces more meaningful results in less time. Optimal query plan provides significant results and also reduces the query processing time. These significant results are large Resource Description Framework graphs.

2.2 Motivation

In today's world we access the Web for many needs. The Web is a collection of Linked data spread over different sources. If a user searches the Web with a simple keyword, it searches for the same across different sources and produces a large number of suggestions, of which many are not relevant to the users need. This process also implies a lot of cost in terms of time and searching. If we build a proper keyword query routing mechanism, we can improve the response time of the query and eliminate most of the suggestions that are not relevant to the keyword.
Fig 2.1 shows the survey on information level required for different categories of people on a search engine [5]. The type of information required for a graduate student on a keyword varies with information required by an undergraduate student on the same keyword. There are millions of users around the world who search the Web for relevant data. They need an efficient and quickly responding search engine that can satisfy their requirements. By properly routing the keyword queries we can improve the user relevant suggestions. This motivated me to propose an enhanced approach for keyword query routing on a search engine.

Fig 2.1: Search engine use behavior of students and faculty [8]
2.3 Project Scope

Keyword Query Routing is always an interesting study in the domain of data mining. Many of the previous frameworks generates results like uncertain data graphs. Then they apply filters and verification techniques on these uncertain data graphs. These filtering techniques are going to eliminate false sub-trees information and finally produce valid candidate results. The next methods proposed takes these uncertain graphs as input and analyzes them to eliminate the frequent sub-tree patterns, reduces the number of patterns and identifies the support for each pattern. These final patterns are displayed as optimal patterns. Then supporting routing plans were generated and applied on these patterns to analyze top k-routing plan's content. In this method we get the advantage like the reduced number of keyword patterns and displaying the content of top routing plans which is more useful. But the problem with all above approaches is that they all have issues like high maintenance cost and more searching cost.
3. SYSTEM DESIGN AND ARCHITECTURE

3.1 Architecture

Routing keyword to the relevant sources reduces higher processing cost of query on all sources. We propose a novel method to generate top k-routing plans that contain the requested query keyword. Unlike the existing system that employs the binary relationship between keywords where there will be a lot of false sub-trees, we employ element level relationship by developing a graph between the keywords at elementary levels. Fig 3.1 gives a high-level overview of the interrelationships between elements at different level and search space in general. Keywords mentioned in any entity description at the element level are linked with set-level element with a relation like type. The set-level elements are present in the sources. Thus there will be an advantage if both the queried keyword elements are connected via a path. A correct routing plan is then selected based on the graphs generated based on relationships between keywords present in the query. This relationships are considered at different levels like element level, set level or keyword level. The final goal is to generate a plan that can search for the keyword from multiple sources and produce the most relevant results.

In record recovery, numerous question extension strategies are taking into account data contained in the top-positioned recovered records in reaction to the unique client inquiry [10], [11],[12]. Essentially, our methodology is in light of performing a starting recovery of assets agreeing to the unique keyword query. Thereafter, further assets will be inferred by utilizing the first recovered ones.

Fig 3.1 shows how elements in a multilevel interrelation graph are connected. Keywords are mentioned in the entity description at the elementary level, these entities are associated with
set-level element via type such as a relation like the object in a Resource Description Framework triple and the set-level elements, which are the relevant results for the keyword, are in the source. The objects related to the subject of the RDF are present in the third layer. These layers form the complete RDF triple which is in the datasets. This RDF triple from the dataset is linked with the sources on the Web that has data of the queried keyword.

![Multi-Level interrelation graph]

**Fig 3.1 Multi-Level interrelation graph**

The main objective of this project is routing a keyword query to produce most relevant results by generating proper routing plans based on the relationship between the subject and object in the RDF triples of the data. In the existing search all the triples are independent and not connected. Hence when a keyword is queried many independent sub-graphs for the keyword are
generated. When there are many sources with same relation but different description i.e., object then many of the sub-graphs are redundant. The quantity of this subgraphs may increase exponentially with the number of sources and the links connecting them. For example, a person called John McCarthy might have multiple awards. Here John McCarthy acts as subject and the different awards are objects and ‘hasawarded‘. In existing search every triple containing John McCarthy as subject and having different awards as objects are all independent and can be assumed as different sub-graphs. In existing search, a routing plan is then applied on the produced subgraphs. This increases the cost of the search and time taken for the search.

Routing keywords to relevant sources may yield better results. For this purpose, in this project a system is developed that when a keyword is queried, all the results for the relevant keywords with same relation are connected to form a multi-level interrelation graph. Instead of producing many independent sub-graphs as in existing search, this system produces a graph that has the related objects with same relation and different description or subject connected. Thus it reduces the number of sub graphs and produces multiple RDF triples. At this step we get RDF triples that has the subject as the query keyword but different relationships. We then apply the routing plans to produce results for the queried keyword. But all the results obtained may not be relevant to the user as the sources containing the queried keyword may all not be relevant to the user. So we then mine the sources obtained as results from the above step and provide a rating for each result based on the number of times the queried keyword appears in the description of the triple. The sources that has the keyword with a user specified rating can only be taken as relevant sources and the results that has least rating are discarded.
Fig 3.2 shows the architecture of the proposed system. The system mainly has two controls: the user control and the admin control. The admin control is mainly used to add data to different tables and add triple information about the data. The system uses oracle 10g as its database to store the data. The data is stored in the form Resource Description Framework (RDF) triples in the database. The Resource Description Framework stores data in the form of subject-predicate-object triples where subject can be the keyword, object is the source of the subject where the predicate defines the relation between the subject and the object. The data is stored in the database in the form of Resource Description Framework triples.

Fig. 3.2 Architecture of the proposed system
The admin login is used to perform admin functionalities. The admin is responsible for entering the data into different tables in the database. In this project the data is entered annually into four different tables which are considered as four different datasets. The admin is also responsible for maintaining the metadata about the datasets of the database. The metadata of the four tables is maintained in another table called triple table. The triple table holds the triple information of all the records present in the tables.

The User login has the search options. When we give a search keyword to the existing search, triples for the linked sources of the queried keyword are retrieved. Then from the triples generated for the linked resources in the existing search we show how the routing plans are generated and display the final triple sequence. The triples generated in the existing search are all individual RDF triples. Then we query the same keyword using the proposed search method. The proposed search produces the triples for the queried keyword in the form multi-level interrelation graph. Then we generate the multiple RDF triples of these triples and generate the final triple sequence. The subjects from the final triple sequences are then fed as search queries for the enhanced search. The enhanced search mines the results which are relevant to the queried keyword that are obtained from the proposed search and calculates the estimations of the keyword in the produced result. In the estimated search the maximum likelihood algorithm is implemented to find the estimations of the queried keyword in the resultant sources. The results in the enhanced search are displayed along with these estimations to show the relevance of the results. Using the filter on the estimations in the enhanced search, the user can view the top-rated results.
Fig 3.3 is the flow chart of the data flow in different searches of the system.

Fig.3.3 Flow Chart of the Search
3.2 System Requirements

The following are the functionalities of this project which are to be implemented to accomplish the objective of the project.

3.2.1 Functional Requirements

- Enter the Query
- Display the Search Results
- Display the selected Query Results
- Display Routing Results
- Triple sequence Results
- Count Values of Triples
- Display Multiple RDF Results
- Display Estimations of Results
- Display the Graph

3.2.2 Software Requirements

The following software’s are used to develop the project.

- Language : Core Java
- Version : JDK 1.7
- IDE : My Eclipse 8.6
• Back-end : Oracle 10g XE

### 3.2.3 Hardware Requirements

The following are the minimal hardware requirements to deploy the application.

- Processor : PENTIUM IV
- Clock Speed : 2.5 GHZ
- Ram Capacity : 2 GB
- Hard Disk Drive : 250 GB
4. System Implementation

4.1 Environment

The proposed system is implemented using Java in My Eclipse 8.6 integrated development environment. Java Server Pages (JSP) is used for the user interface of the project. The server used is Apache Tomcat 7.0 and the database used to store and retrieve the data is Oracle 10g Express edition.

4.1.1 Java Server Pages (JSP)

Java Server Pages (JSP) is a technology widely used for controlling the content or appearance of the Web pages through the utilization of servlets, little projects that are indicated in the Web page and run on the Web server to alter the Web page before it is sent to the client who requested it. Sun Microsystems, the designer of Java, additionally alludes to the JSP innovation as the Servlet application program interface (API) [16]. The main functionality of a Java Server Page is to call a Java program that is executed by the Web server, an Active Server Page contains a script that is interpreted by a script interpreter such as JScript before returning the page to the user.

4.1.2 Java Development Kit (JDK)

JDK is released by Oracle Corporation. It is a software development kit (SDK) containing tools for developing JAVA based applications. It includes the Java Runtime Environment (JRE) and other tools needed for developing, debugging and monitoring java applications.

4.1.3 Java Springs

Spring framework was initially written by Rod Johnson and was first released under the Apache 2.0 license. Java springs is an open source platform that supports developing applications
using Java very easily and rapidly. Spring is the most popular application development framework used to create high performing, easily testable and reusable code.

4.1.4 MyEclipse

MyEclipse is an integrated development environment (IDE) that is mainly used for Java. MyEclipse unify all Java EE technologies under a single stack that supports almost everything. It has application servers like tomcat which is used to deploy this project.

4.2 Web of Data

A graph-based data model is built to represent different data sources. We consider different tables in the database as different data sources. Element-level data graph is used to represent relation between individual data elements and Set-level data graph represents the relation between a groups of elements. In this model the tables in relational database are assumed to be RDF resources, the values as literals of RDF and the relations and attribute values are assumed to be RDF triples. The tuple in the database is considered as entity and the inter entity relations are built using foreign key relations. Thus a Web of data is built that is similar to the data on the Web and linked using Resource Description Framework. We built a Web of data using four tables. The Employee table has the information about the employment of people, the Award table has the information about various awards received by people, Sports table has information about sports men and the books table has information about information about authors of the book. We maintain a table called Triple which has the metadata of all the four tables.

4.3 User Interface

The user interface has the login access to the users and the administrator. The administrator has the access to update the data into different datasets in the database insert or delete data from
the database and create new data sets. The admin updates the triples information of the data from different datasets.

The user login has the search bar where the user can query a keyword and view the relevant results in the existing search, proposed search and the enhanced searches. The time taken for each search is fed into a graph chart that analyzes the time taken for each search. The user login also displays the triples, routing plans generated for each search.

4.3.1 Existing Search

When we enter the query keyword in the search bar of the existing search, it searches for the relevant data in different data sets and produces the results. In the existing search, the system primarily searches at the element level and forms subgraphs that has the keyword and the elements that are related with the keyword at element level and then searches for the relevant results in the set level graph. In existing search the only apart of the database is queried and no external edges are formed. When searched for a keyword john the existing search first searches for the john in element level and finds the elements that are related to it at element level and then generates an element level graph and then searches for the relevant data in the set level graph formed by connecting all the sources that refer john. Then this set-level graph is searched to find the relevant results for the queried keyword. Then a top-k routing plan is generated to find the relevant results.

In Fig.4.1, we query a keyword ‘Edison’ in the existing search.
Fig. 4.1: Searching for the keyword “edison” in Existing Search

Fig 4.2 shows the displayed results for the queried keyword ‘edison’. There are total 6 records matching for the keyword. Out of the 6 results displayed, 5 records are having information about Thomas Alva Edison and one record has information about Dr. Edison Rodrigues. We now show the triples generated for the Thomas Alva Edison.

**Search Results**

![Image of search results]

**Fig. 4.2 Search results for the keyword Edison in Existing Search**
To fetch the resultant records from different datasets, first the metadata table is searched for the subjects matching with the keyword Thomas Alva Edison. Fig 4.3 shows the triples generated for the queried keyword.

**View Triples**

![Triples generated for the keyword “Thomas Alva Edison”](image)

The triples from the different datasets are linked and a count value is generated for each connection. The count value is generated by dividing 1 with total number of triples and multiplying...
the value with number of connected triples. The triple sequence with count value 1 gives the final triple sequence which retrieves the records for the queried keyword. Fig. 4.4 shows the final triple sequence generated for the keyword ‘edison’.

Final Triples Sequence

<table>
<thead>
<tr>
<th>Final Triples Sequence</th>
<th>CountValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>[{\text{Thomas Edison, hasInvented, Phonograph,}}, ]</td>
<td>0.1666666666666666</td>
</tr>
<tr>
<td>[{\text{hasInvented, Bulb,}}, {\text{Thomas Edison, hasInvented, Phonograph,}}]</td>
<td>0.3333333333333333</td>
</tr>
<tr>
<td>[{\text{Thomas Alva Edison, wasAwarded, Officer of the Legion of Honour,}}, ]</td>
<td>0.5</td>
</tr>
<tr>
<td>[{\text{hasInvented, Bulb,}}, {\text{Thomas Edison, hasInvented, Phonograph,}}]</td>
<td>0.6666666666666666</td>
</tr>
<tr>
<td>[{\text{ives, Milan,}}, {\text{Thomas Edison, wasAwarded, Matteucci Medal,}},]</td>
<td>0.8333333333333334</td>
</tr>
<tr>
<td>[{\text{Thomas Alva Edison, wasAwarded, Officer of the Legion of Honour,}},]</td>
<td>1.0</td>
</tr>
<tr>
<td>[{\text{hasInvented, Bulb,}}, {\text{Thomas Edison, hasInvented, Phonograph,}}]</td>
<td></td>
</tr>
</tbody>
</table>

Fig.4.4 Final triple sequence of keyword “Thomas Alva Edison”

4.3.2 Proposed Search

In the proposed search when a keyword is queried, the system first searches for the element in the elementary level and finds the relevant elements and forms an element level graph. Then all the datasets are searched for the sources that contains the attributes for the queried keyword. A set-level is graph is formed at this point that relates all the sources containing the keyword using the
relationships provided in the metadata table. There is a chance of getting incorrect results or no displaying results if the metadata is not maintained properly. Once the set-level subgraph is generated, the elements having same relations with source attribute are grouped together to form subgraphs which makes the search quicker and thereby reducing the cost of the search.

Unlike in the existing search where each triple is connected individually and generating a count value, in the proposed search all the relevant linked resources are extracted at a time and grouped together to form a triple sequence and the triples having similar relation are all grouped together to form the final triple sequence.

Fig.4.5 shows the triples generated in the proposed search. These triples are similar to the triples generated in the existing search.

**View Triples**

| {Thomas Edison, hasInvented, Phonograph} |
| {Thomas Edison, hasInvented, Sub} |
| {Thomas Alva Edison, wasAwarded, Officer of the Legion of Honour} |
| {Thomas Edison, wasAwarded, Matteucci Medal} |
| {Thomas Edison, lives, Milan} |
| {Thomas Edison, wasBorn, Thomas Alva Edison (February 11, 1847 ? October 18, 1931)} |

Fig: 4.3 Triples generated for the keyword “Thomas Alva Edison” in proposed search
These triples from different datasets are then linked together to form a triple sequence that has information about all the linked sources containing data about the queried keyword in the metadata table. Fig 4.4 shows the triple sequence of the queried keyword in the proposed search.

**Triples Sequence**

![Triples Sequence](image)

**Fig.4.5 Triple sequence generated for the keyword “Thomas Edison” in proposed search**

The triples having same predicate or similar relation are them grouped together to form the final triple sequence. The final triple sequence has reduced number of triples and it makes the search faster to retrieve results. Fig 4.7 shows the final triple sequence generated for the queried keyword.
Final Triples Sequence

Final Triples Sequence

Fig: 4.6 Final Triples sequences generated in proposed search

4.3.3 Enhanced Search

The subject elements from the triples generated in the proposed search are given as input to the enhanced search. The enhanced search applies the concepts of Maximum Likelihood Algorithm on the resultant linked resources of the proposed search to get the estimations of the keyword in result sources. With obtained estimation values we get the knowledge of how relevant are the resulted sources to the user query. Fig 4.7 shows the querying of the enhanced search. In enhanced search the subject part of the triples generated are all extracted and given as input to the enhanced search. The enhanced search then searches different datasets to retrieve matching results for the queried keyword.
Fig 4.7 Querying the Enhanced Search

The enhanced search then mines the linked resources to find the number of times the queried keyword is present in the sources. It mines the entire result document to find the count of the occurrence of the keyword in the result document which is displayed along with the results. With the count we can determine how relevant is the document for the user query and can filter the documents that are less relevant by setting a count threshold for the result to be displayed. Fig 4.8 shows the results displayed in the enhanced search with the occurrence count value of the queried keyword.

Fig. 4.8 Results generated with count values
To display only the results that are most relevant, a threshold can be applied on the count values. When we apply a filter on the results obtained in the above step to get only the results with count values of 3 or more, the records with count values less than 3 are filtered. Thus only records with high relevance can be obtained. Fig 4.9 shows the results after applying the filter on the results of the queried keyword.

**Search Results**

<table>
<thead>
<tr>
<th>Thomas Edison - Biography - Inventor - Biography</th>
<th>3 times</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.biography.com/people/thomas-edison-9284349">http://www.biography.com/people/thomas-edison-9284349</a></td>
<td></td>
</tr>
<tr>
<td>Thomas Edison had acquired a reputation as a first-rate inventor. Thomas Edison set up his first small laboratory and manufacturing facility in Newark. Edison formed numerous partnerships.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>The Inventions of Thomas Edison - Inventors - About.com</th>
<th>4 times</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://inventors.about.com/library/inventors/bl">http://inventors.about.com/library/inventors/bl</a> Edison.htm</td>
<td></td>
</tr>
<tr>
<td>Thomas Edison established the Edison Speaking Phonograph Company. The first great invention developed by Thomas Edison in Menlo Park is the foil phonograph. Thomas Edison, Thomas Edison</td>
<td></td>
</tr>
</tbody>
</table>

Fig 4.9 Results after applying a threshold on the results.
5. TESTING AND EVALUATION

Process of finding differences between the expected behaviors specified by system models and the observed behavior implemented system is called “System testing” [19]. The goal of testing is to design test cases that exercise defects in the system and to reveal problems. Testing is process of finding bugs in the system that might have crept in during the various phases of the project development starting from design phase to implementation by designing efficient test cases. An efficient set of test cases is the first step in finding the deviations of the system from expected behavior. The quality of a product can be judged by testing.

5.1 Unit Testing

In unit testing every single unit of the system or a group of units are tested. It is tested for the expected outputs for given inputs of the unit that is implemented [19]. It is usually done by the developers of the product. The main testing involves determining whether the presented algorithm is operating properly by making different types of input to the scheme. In this system, search results for each search is tested for the correctness of the result and whether the each component of the system is functioning as designed or not.

5.2 System Testing

Determining whether the system developed that is being delivered is meeting the scope of the project and its specifications and its objectives [20] is performed in the system testing. This is usually done before the product goes into the production phase. Testing is usually of two types alpha testing and beta testing. In alpha testing, the testing is performed by the developers of the product before it is shipped to the end users for further testing and production stage.
In beta testing, the end users have the hands on experience with the developed product to be tested and make a note of all the bugs they are experiencing while using the product. The system developed in this project is tested by testing the relevant results of the search queries. The results obtained must be relevant to the search query and the cost of the proposed search must be less than the existing search.

5.3 Test Cases

Test cases are the testing scenarios developed by the developer to verify whether the system is performing as designed for various inputs and producing the expected outputs.

Test case 1: Login

This system has two logins: the Admin Login and the User login. In this test case we test whether the admin can login and update the data into various data sets. In the user login we verify whether a registered user can login to search for his keyword in the system.

Test case 2: Querying Existing search

When the user provides a keyword as a query in the existing search, the system first searches for the keyword element in different data sets and forms an element level sub-graph and then with the subsets formed from the elementary searches for the sources containing the keyword to form a set level graph. Then a top-k routing plan is generated to search for the queried keyword.
When we search for a keyword Sachin, the system searches for the metadata table for elements containing Sachin in the subject part of the triple. The existing search produces results having Sachin in the subject. It produced a total of 33 records having Sachin as a part of the subject. Then we select Sachin Tendulkar as the keyword Query. Fig.5.1 shows the results displayed for the query “Sachin Tendulkar”. There are total 10 results generated for the query. These 10 records are from the four datasets we created for this project.

<table>
<thead>
<tr>
<th>Sachin Tendulkar - Wikipedia, the free encyclopedia</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.cricketrun.com">www.cricketrun.com</a></td>
</tr>
<tr>
<td>Sachin Tendulkar named after his favourite music director, Sachin Dev Burman. Tendulkar has three elder siblings: two half-brothers Nihin and Ajiit, and a ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sachin Tendulkar - ESPN Cricinfo</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.espn.cricinfo.com/india/content/player/35320.html">www.espn.cricinfo.com/india/content/player/35320.html</a></td>
</tr>
<tr>
<td>In a nutshell Perhaps the most complete batsman and the most worshipped cricketer in the world, Sachin Tendulkar holds just about every batting record worth owning ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sachin Tendulkar</th>
<th>Facebook</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://www.facebook.com/SachinTendulkar">https://www.facebook.com/SachinTendulkar</a></td>
<td></td>
</tr>
<tr>
<td>Sachin Tendulkar: 22418113 likes □ 744911 talking about this. The Official facebook page of Sachin Tendulkar managed by Seven3Rockers ....</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sachin Tendulkar - CricketArchive</th>
</tr>
</thead>
<tbody>
<tr>
<td>cricketarchive.com/Archive/Players/1933/1933.html</td>
</tr>
<tr>
<td>Full name: Sachin Tendulkar. Born: 24th April 1973, Bombay (now Mumbai), Maharashtra, India. Batting: Right-hand batsman. Bowling: Leg-break ...</td>
</tr>
</tbody>
</table>

**Fig 5.1: Search results for the keyword “Sachin”**

The triples generated for the query are shown in Fig 5.2. There are total 10 triples retrieved for the keyword.
Fig 5.2: Triples generated for the keyword “Sachin”

Then these triples are connected and the count value is calculated. Then the triples with same relation are connected and a final triple sequence is generated. This final triple sequence is used to retrieve results from the different datasets. Fig 5.3 shows the final triple sequence with generated count value.
Test case 3: Querying Proposed Search

When a keyword is queried in the proposed search, the system searches the metadata table for the linked resources and generates the triples of the records having the keyword in the subject. Fig 5.4 shows the triples generated for the queried keyword in the proposed search.
Fig: 5.4 Triples generated in the Proposed Search

The triples are then linked which can be assumed as a graph. The triples from each dataset are grouped first and then the triples from all the datasets are grouped to generate the triple sequence. Fig 5.5 shows the triple sequence generated for the keyword “Sachin”.

Fig: 5.5 Triples sequence generated in the Proposed Search for the keyword “Sachin”
The triples are then mined to get the triples having same predicate or relation. The triples with similar relation are then grouped together that with one subject and the different objects connected to form a final triple sequence. Fig 5.6 shows the final triple sequence which is used to retrieved results from the linked resources containing data of the queried keyword.

![Final Triples Sequence](image)

Fig: 5.6 Final Triples sequence generated in the Proposed Search for the keyword “Sachin”

Test case 3: Querying Enhanced search

Enhanced search takes the subject part of the triples from the proposed search as input and retrieves results from the linked resources along with the occurrence of the queried keyword in the result document. This count value shows the relevance of the result document to queried keyword.
Fig 5.7 shows the querying of the enhanced search. The enhanced search automatically takes the subject part of the triples from proposed search as input query.

Fig. 5.7 Querying the Enhanced Search

There are total 10 results displayed with the count values displayed with them. The count value determines the relevance of the results produced. Fig 5.8 shows the results with count values.

Fig. 5.8 Search results with estimation values in Enhanced search
The results produced in the enhanced search can be filtered by giving a threshold value to the count obtained in the above step. When a threshold value of 3 is given to the results in the above step, all the results with count value less than 3 are filtered and the records with the count value 3 or more are displayed. The total number of records displayed came to 5 from 10. Fig 5.9 shows the records that have count value 3 or more.

**Search Results**

![Search Results](image)

*Fig. 5.9 Search results with count values in Enhanced Search for the keyword “Sachin” after applying filter*

**Test Case 4: Displaying the histogram**

The time taken for each search is calculated from the session time of each search and are fed as inputs to a histogram. Time is taken as the co-ordinate on the Y-axis and each search is taken as co-ordinate on the X-axis. The graph chart produced shows the time taken for each search.
In the testing phase, this project is queried with a keyword Sachin. The database has a total of 33 records with Sachin as subject in their triples. Then the query is mapped with an element Sachin Tendulkar. This keyword has a total of 10 records in the four different datasets of the database. The existing searched produced the result in 22.86 milliseconds, the proposed search took approximately 12 milliseconds to retrieve the records whereas the enhanced search after getting the triples from the proposed search took 6.85 milliseconds.
6. CONCLUSION

This project proposes the idea of routing keyword query to produce more relevant results by implementing relationship graphs between the keywords at different levels. This idea proposes to reduce the high cost of searching for structured data spanning across multiple resources by routing the keywords only to the relevant sources. A correct routing plan will be selected by using graphs developed based on the relationships between keywords in the query at different level. This project is tested with a database having 856 records in four different datasets. The records in the datasets are created such that each dataset will some kind of data about the keywords. By this we created a web of data similar to the Linked data on the internet where information about a keyword may encompasses on different sources.

Queries with more keywords would also generate effective results, but they cannot be handled efficiently. For example, if we give a query with more keywords as a query in the existing system, it would also give effective results, but it might take a higher time which is not desirable in a present day's demand of high responsiveness. Keyword search without routing is problematic when the query has many words. That is the reason for routing of queries having more number of keywords.
Bibliography and References


[5]. Oya Y. Rieger, "Search engine use behavior of students and faculty: User perceptions and implications for future research"


[9]. Thanh Tran and Lei Zhang, “Keyword Query Routing IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 2, FEBRUARY 2014.”


