Logical Schema-Based Mapping Technique to Reduce Search Space in the Data Warehouse for Keyword-Based Search

Fiaz Majeed and Muhammad Shoaib
Department of Computer Science and Engineering, University of Engineering and Technology, Pakistan

Abstract: *Data warehouse systems are used for decision-making purposes. The Online Analytical Processing (OLAP) tools are commonly used to query and analysis of results on such systems. It is complex task for non-technical users (executives, managers etc.,) to query the data warehouse using OLAP tool keeping in view the schema knowledge. For such data warehouse users, a natural language interface is a viable solution that transparently access data to fulfil their requirement. As data warehouse contain several times more data (that increase with incremental refreshes) than the operational systems. So keyword-based searching in such systems cannot be performed similar to database based natural language systems. Existing natural language interfaces to data warehouse commonly explore keywords in data instances directly that takes more than sufficient time in generating results. This paper proposes a Logical Schema-based Mapping (LSM) technique to reduce search space in the data warehouse data instances. It performs mapping of the natural language query keywords with logical schema of the data warehouse to identify the elements prior to search in the data instances. The retrieved matches for a keyword are ranked based on six criteria proposed in this paper. Further, an algorithm has been presented which is developed upon the proposed criteria. Targeted search in the data instances is then performed efficiently after the identification of schema elements. The in-depth experiments have been carried out on real dataset to evaluate the system with respect to completeness, accuracy and performance parameters. The results show that LSM technique outperforms the existing systems.*

Keywords: *Database systems, natural language processing for data warehouse, information systems, data warehousing, natural language interface, keyword-based query processing.*

Received January 22, 2014; accepted October 14, 2014

1. Introduction

Data warehouse is used by executives for decision-making to boost up their business [6, 7]. It maintains historic data to answer user queries. In comparison to Online Transaction Processing (OLTP) systems, data warehouse semantics completely differentiate it. Thus, it uses tools and technologies to make it knowledge supporting system rather than transaction processing. In this regard, natural language interfaces for OLTP systems cannot be used for Online Analytical Processing (OLAP) supported systems. Research community is focusing to build natural language interface to data warehouse keeping in view its semantics.

The Natural Language Interface to Database (NLIDB) facilitates users to write query in natural language without having knowledge about schema and technical query language. Several NLIDB's have been developed. Classical works including [1, 3, 9, 13, 15] provide technical solutions that map natural language query accurately in the structured databases [8, 15, 16, 23, 25]. Besides other query tools [2], research community has started efforts to build natural language interface to data warehouse. According to best of our knowledge, there is limited number of natural language interfaces to data warehouse. Existing systems provide solutions at very initial level. In Keyword-Driven Analytical Processing (KDAP), analysis is not performed from performance point of view, OLAP engine is not utilized, it requires more than sufficient user interaction, and it does not rank measures [24]. The Question Answering (Q and A) system addresses working on aggregation queries but slice and dice and drill-down queries are not addressed, accuracy and execution time analysis is not performed and results are not compared with existing systems [18]. In Query Generator (QueGen), solution is not provided if a keyword does not match with any element of the schema, ambiguity in mapping the value, build incomplete query and accuracy has not been measured [21].

In this paper, Logical Schema-based Mapping (LSM) technique has been presented to enhance performance and accuracy at early stage. At first step, logical schema is used to map the keywords generated from the natural language query. In this way, technical elements are identified before searching in the data instances thus reduces search space. The contributions of this paper are as follows:

- It presents the LSM technique to identify target elements (dimension table names, dimension attributes, fact table names and measures) before searching into the data warehouse data instances. The data instances are then explored for already identified elements. So this technique reduces search space in the data instances and increase significant performance at the moment of searching.
- To rank the multiple matches for a keyword, six criteria have been presented. Based on the criteria, a ranking algorithm has been developed.
- It focuses on the performance of keyword-based processing in comparison to earlier works on the data warehouse.
- The data instances are searched in already identified elements rather than greedy search which is adopted by existing works.

The paper is organized as follows: Section 2 provides the related work. The proposed solution is discussed in section 3. Further, section 4 presents the natural language query keywords mapping and ranking criteria whereas, searching strategy for the data instances is given in section 5. Later, evaluation of the proposed solution has been carried out in section 6. Finally, conclusions and future directions are presented in section 7.

2. Related Works

A limited number of natural language interfaces to data warehouse have been appeared in the literature [20]. The KDAP [24] is one of such systems. It takes natural language query from the user and divides into keywords. Depending upon different interpretations of a keyword, a number of join paths termed as starnets are generated. KDAP directly search in the data instances using full-text indexes and use greedy search approach. It forms join paths using logical schema after searching of the desired values. So ranking of the starnets is performed at data instances level. The Q and A system [18] focuses on natural language query completion before its execution on the data warehouse. It provides framework for natural language query processing. The framework focuses on parsing of the query. The query parsing in this way without analysis of data is prone to ambiguities. This approach search keywords in the domain thesaurus built based on the logical schema and the data contained in the dimensions. Further, it does not rank retrieved occurrences. It only searches new terms specified in the user query, which do not found in the domain thesaurus, find and rank from the web ontology. It finally generates an OLAP query. To formulate OLAP queries from the natural language input; a tool QueGen [21] has been proposed. It semantically parses the natural language query and maps to the OLAP query. The QueGen generates query by searching in the data instances directly and search elements in a particular sequence (i.e., tables, fields, functions and field values). It is Natural Language Processing (NLP) based approach that translates query semantically.

Several NLIDB's have been proposed that represent schema in graph form. Resulting graph is then used to translate natural language to structured query. The KDAP [24] uses a breadth-first approach to find join-paths for the user query. Discover [13] search all Candidate Networks (CNs) from the schema graph. The CNs are then evaluated from which common sub-expressions are identified and minimum CNs is chosen. Bi-directional search [17] is carried out in backward and forward directions. Initially, a keyword is searched in the graph that has minimum number of matching nodes. Backward search is carried out through such keyword while forward search is performed through potential roots. Thus, search is narrowed to find answer tree efficiently. ME/R model for data warehouse is presented in [22].

To order multiple interpretations of a keyword, the work presented in XRank [11] assigns weight to each node and generates a result tree. Keyword proximity search [10] proposes an engine that forms answer set and rank those answers based on relevancy. The ranking [5, 12, 19] is performed to select relevant answers from large number of answer set. Ranking measures compute close connection among keywords and weight of the answer. XML ranking is also carried out in [4]. It identifies the user intention based on statistics and ranks the result. XKeyword [14] provides proximity search on XML schema. It computes all CNs.

3. Proposed Solution

3.1. Formal Definition of the Problem

The logical schema can be modelled as a directed graph. The nodes of the graph are entities demonstrating dimension and fact tables. An edge symbolizes the relationship among the fact-dimension or dimension-dimension table. The edges going out of the fact table(s) attached with the dimensions which are called atomic dimensions (or level 1 dimensions). Further, edges going out of the atomic dimensions are drill-down levels. As levels increase, data is further drill-down. Conversely, if levels decrease, data is rolled up. Each fact table contains foreign key attributes of atomic dimensions. It is also composed of measures upon which aggregations are calculated. Further, a dimension is composed of a set of attributes including a primary key.

In this approach, we model the logical schema as a graph G (V, E) such as V=D U F where D denote the set of dimensions whereas F denote the set of fact tables in the schema. Each di=(a_{pk}, A) is a set of attributes A={a_1, a_2, a_3, …, a_n} and a primary key attribute a_{pk}. Further, each f_i=(a_{fk}, M) is the set of measures M={m_1, m_2, …, m_n} and composite foreign key attribute a_{fk}. The relationship is defined as $d_i(a_{pk})=f_i(a_{fk})$. This expression shows the relationship among each dimension and fact table. Whereas, $d_i(a_{pk})=d_i(a_{fk})$ is termed as the drill-down level which is related through two dimensions. The drill-down levels are denoted as L={l_1, l_2, l_3, …, l_n} where $l_1=d_1(a_{pk})=d_2(a_{fk})$, $l_2=d_2(a_{pk})=d_3(a_{fk})$ and so on.

The formal definition of the natural language query mapping is given as: Let Q be the natural language user query which is composed of K keywords. Each keyword k_i can be mapped to D_i, F_i, A_i, or M_i. Finally, a list of schema elements is retrieved in the result of mapping of a query Q.

3.2. Solution Architecture

The solution architecture is depicted in Figure 1. It consists of the following components:

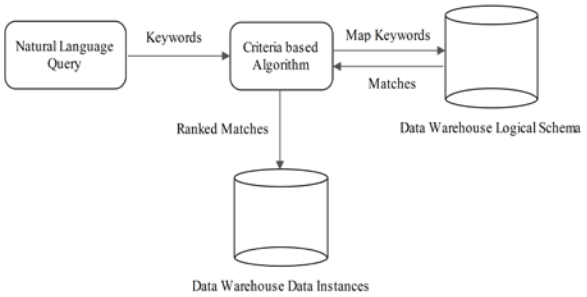


Figure 1. Architecture of the LSM technique.

- Natural Language Query: Is the user input which is composed of keywords. Such keywords are generated by applying the IR method i.e., the removal of stop words and wh-word (Select, display, show etc., are wh-words which cannot present retrieval criteria) from the user input. To retrieve non-trivial results, these must be relevant to the domain.
- Criteria-Based Algorithm: Has been developed based on six criteria discussed in section 4. It retrieves all element matches for each keyword and ranks them.
- Data Warehouse Logical Schema: Is composed of the elements (dimension table names, dimension attributes, fact table names and measures). The algorithm maps the keywords in the logical schema and retrieves element matches.
- Data Warehouse Data Instances: Contain physical records related to dimensions and fact tables. As criteria-based algorithm rank matches, those are then explored for values search, specified in the natural language user query.

4. Natural Language Query Keywords Mapping

A list of keywords is generated from the natural language user query and for each keyword a list of element matches are retrieved from the logical schema. In open natural language input, user may write any keyword that might not match with required element in the schema but its synonym may return that particular match. Thus, synonym support has been provided in this work to increase the possibility of matching with accurate element in the schema. To fulfil this objective, a schema based domain thesaurus has been built that maintains all relevant synonyms for each element. In addition, it maintains following necessary elements which are not available in the schema but having necessary part of the OLAP query. Such elements include following:

- Aggregation Functions: To identify aggregation demand, a list of aggregation functions with relevant synonyms is maintained. Such aggregation functions include AVG, SUM, MIN, MAX, STDDEV etc.
- Aggregation Levels: The data warehouse schema aggregates the data at hierarchical levels. To rollup up the data at required level, accurate level attribute is necessary to be identified. Such keywords in the query include yearly, quarterly, monthly, weekly and product-wise etc. For example, yearly keyword maps to the attribute CalendarYear of the dimension DimTime.
- Derived Measures: The derived measures are calculated from the existing measures similar to derived attributes. For example profit keyword refers to a derived measure SUM (SalesAmount-UnitPrice).
- Range Elements: Range elements define the range criteria for the selection of values from the dimension tables. The keywords referring to such keywords are also maintained in the domain thesaurus. Such keywords include before, after, between and current etc. The keyword before is mapped with less than "<" operator and after with greater than ">" and so on.
- Criterion 1: At first, match a query keyword with the list of aggregation functions and aggregation levels. If match is found then stop searching for that keyword.
- Formal Definition: A keyword k_i of a query Q (composed of K keywords) is matched in the domain thesaurus. As a result, a list of synonyms S is retrieved including k_i. Each candidate $s_i \in S$ is then matched with a list of aggregation functions AF. If

$$\left(s_i = \varphi af_j\right) \neq \tag{1}$$

Where $af_j \in AF$ and i/j= 1, 2, 3, ..., n.

The match is stored in the list of identified elements. If no match is found in AF, find each s_i in the list of aggregation levels AL. If

$$\left(s_i = \varphi al_j\right) \neq \tag{2}$$

Where $al_j \in AL$.

The match is stored in the list of identified elements.

4.1. Retrieval of Matches and Ranking

If criterion 1 returns ϕ for a keyword then generate its matches (interpretations) by the schema mapping. As all possible matches are retrieved, those are ranked using criterion 2-6. The ranking criteria order the matches according to relevance and set the most relevant element at top.

- Criterion 2: Order the dimensions according to their hierarchical level. Further, arrange attributes/measures under container Dimensions/facts. Group them even if dimension/fact name is not available in the matches.
- Formal Definition: If *MATCH* be the set of matches generated from mapping a keyword with the schema then:

$$Order\left(dim\left(match_i\right)\right) \tag{3}$$

$$\left\{Group\left(dim\left(match_i\right), match_j\right) : attrib\left(match_j, match_i\right)\right\} \tag{4}$$

Where $i = 1, 2, 3, ..., n$ and $j = 1, 2, 3, \ldots, m$.

The Order function in Equation 3 orders dimension elements in hierarchical sequence.

In Equation 4, dimension elements are grouped with its attributes which are represented as *attrib*(*match*$_j$, *match*$_i$).

- *Example* 1: The ranking of matches according to criterion 2 is illustrated in Table 1. The dimension table names are shown in italic whereas their elements have been grouped under them.
- Default Elements: In each dimension and fact table, a default element is designated. These are chosen based on frequency of their usage in the user queries. For instance, EnglishProductName attribute is chosen as default because it is most frequently used in the query set than other attributes of the same dimension.

Table 1. Illustration of ranking according to criterion 2.

Query #	Keyword	Matches
3	Product	DimProduct, EnglishProductName (DimProduct), SpanishProductName (DimProduct), FrenchProductName (DimProduct), ProductLine (DimProduct), DimProductCategory, EnglishProductCategoryName (DimProductCategory), SpanishProductCategoryName (DimProductCategory), FrenchProductCategoryName (DimProductCategory), DimProductSubCategory, EnglishProductSubCategoryName (DimProductSubCategory), SpanishProductSubCategoryName (DimProductSubCategory), FrenchProductSubCategoryName

- Criterion 3: After execution of criterion 2 for a keyword, rank default attributes/measures under their dimensions/ facts name at top.
- Formal Definition: If *default* (*MATCH*) be the list of default matches.

$$\left\{match_i \in default\left(MATCH\right) : top\left(match_i\right)\right\} \tag{5}$$

Where *top*(*match*$_i$) places the default attribute at top under their dimensions or facts.

- *Example* 2: The ranking of matches according to criterion 3 is illustrated in Table 2. Default elements are taken in top order of ranking within their respective group. The matches shown in italic are default attributes and measures.
- Criterion 4: After execution of the criterion 3 for a keyword, further count matches of each element type (i.e., dimension or fact) and rank the fact or dimension elements based on maximum count.

Table 2. Illustration of ranking according to criterion 3.

Query #	Keyword	Matches (Post execution of Criterion 3)
10	Sale	DimSalesReason SalesReasonName (DimSalesReason) SalesReasonType (DimSalesReason) DimSalesTerritory SalesTerritoryRegion(DimSalesTerritory) SalesTerritoryCountry(DimSalesTerritory) SalesTerritoryGroup(DimSalesTerritory) FactInternetSales SalesAmount (FactInternetSales) SalesOrderNumber (FactInternetSales) SalesOrderLineNumber (FactInternetSales) FactInternetSalesReason FactResellerSales SalesAmount (FactResellerSales) SalesOrderNumber (FactResellerSales) SalesOrderLineNumber (FactResellerSales) FactSalesQuota SalesAmountQuota (FactSalesQuota) SalesPersonFlag (DimEmployee) AnnualSales (DimReseller)

- Formal Definition:

$$Max\begin{pmatrix} count\left(dim\left(match_i\right) + attrib\left(match_j, match_i\right)\right), \\ count\left(fact\left(match_i\right) + measure\left(match_j, match_i\right)\right) \end{pmatrix} \tag{6}$$

Max() function in Equation 6 returns maximum count of elements type i.e., dimension or fact.

- *Example* 3: The ranking of matches according to criterion 4 is illustrated in Table 3. As number of the fact elements are greater than the dimension elements so those are ranked at top.

In query 10, total 11 matches belongs to fact tables and remaining 9 are from the dimensions. So fact matches are ranked at top.

Table 3. Illustration of ranking according to criterion 4.

Query #	Keyword	Matches (Post execution of Criterion 4)
10	Sale	FactInternetSales SalesAmount (FactInternetSales) SalesOrderNumber (FactInternetSales) SalesOrderLineNumber (FactInternetSales) FactInternetSalesReason FactResellerSales SalesAmount (FactResellerSales) SalesOrderNumber (FactResellerSales) SalesOrderLineNumber (FactResellerSales) FactSalesQuota SalesAmountQuota (FactSalesQuota) DimSalesReason SalesReasonName (DimSalesReason) SalesReasonType (DimSalesReason) DimSalesTerritory SalesTerritoryRegion(DimSalesTerritory) SalesTerritoryCountry(DimSalesTerritory) SalesTerritoryGroup(DimSalesTerritory) SalesPersonFlag (DimEmployee) AnnualSales (DimReseller)

- Criterion 5: The non-confirmed dimensions in the matches make the selection of their corresponding fact table unique.
- Formal Definition: If weight of each outgoing link of a dimension table with fact table is equals to 1 then sum of unique links is always 1.

$$\sum_{j=1}^{n} dim \rightarrow f_j = 1 \tag{7}$$

- *Example* 4: The ranking of matches according to criterion 5 is illustrated in Table 4. For instance DimEmployee is the non-conformed dimension of the FactResellerSales so elements of other fact tables have been removed from the matches list.

Table 4. Illustration of ranking according to criterion 5.

Query #	Keyword	Matches (Post execution of Criterion 5)
18	Sale	FactResellerSales SalesAmount (FactResellerSales) SalesOrderNumber (FactResellerSales) SalesOrderLineNumber (FactResellerSales) FactSalesQuota SalesAmountQuota (FactSalesQuota) DimSalesReason SalesReasonName (DimSalesReason) SalesReasonType (DimSalesReason) DimSalesTerritory SalesTerritoryRegion(DimSalesTerritory) SalesTerritoryCountry(DimSalesTerritory) SalesTerritoryGroup(DimSalesTerritory) SalesPersonFlag (DimEmployee) AnnualSales (DimReseller)
	Employee	DimEmployee NumberEmployees (DimReseller)

- Criterion 6: Two keywords can be mapped into a single element if both have same element in their respective matches list.
- Formal Definition: If k_i and k_j be two query keywords such that:

$$k_i + k_j \approx match_k \tag{8}$$

Where k_i and k_j are mapped to $match_k$.

- Example 5: The ranking of matches according to criterion 6 is illustrated in Table 5. As DiscountAmount (FactResellerSales) and DiscountAmount (FactInternetSales) are available in both 'Discount' and 'amount' keywords, thus both represent the same element.

Here, 'Discount' and 'amount' keywords both have common measure DiscountAmount in their matches. So these represent the same element.

Table 5. Illustration of ranking according to criterion 6.

Query #	Keyword	Matches
5	Discount	DiscountPct (DimPromotion) UnitPriceDiscountPct (FactInternetSales) DiscountAmount (FactInternetSales) UnitPriceDiscountPct (FactResellerSales) DiscountAmount (FactResellerSales)
	Amount	Amount (FactFinance) ExtendedAmount (FactInternetSales) DiscountAmount (FactInternetSales) SalesAmount (FactInternetSales) ExtendedAmount (FactResellerSales) DiscountAmount (FactResellerSales) SalesAmount (FactResellerSales) SalesAmountQuota (FactSalesQuota)

4.2. Ranking Algorithm

The algorithm is based on six criteria discussed above. It evaluates each criterion in sequence (criterion 1-6) on the matches for a keyword to identify accurate element. It is presented in Algorithm 1.

Algorithm 1: Ranking multiple matches for a keyword.

```
Input: synonyms S (including k).
Output: Ordered Elements.
Method:
If $(s_i = af_j) \neq \varphi$ OR $(s_i = al_j) \neq \varphi$ Then
    Return
  End If
  Generate matches for each $s_i$
  Order dimensions dim(match$_i$)
  Group attributes with their container dim(match$_i$)
  If match$_i \in$ defult (MATCH) Then
    Place match$_i$ at top than other attributes of container dim(match$_i$)
  End If
(Count (dim elements) > Count (fact elements)? Top (dimension elements): Top (fact elements))
If $\sum_{j=1}^{n} \dim \rightarrow f_j = 1$ then
   Eliminate other fact matches from the matches list
End If
If $k_i + k_j \approx match_k$ Then
  $k_i + k_j$
End If
```

The detail of the algorithm is as follows: The keyword and its synonyms are searched in the repository of aggregation functions and aggregation levels. If any match is found, algorithm stops from further processing (line 1-3). Otherwise all matches for the keyword and its synonyms are retrieved with schema mapping (line 4). The retrieved matches are then ordered in the way; the dimensions/facts and their attributes/measures are grouped together (line 5-6).

In continuation of previous ordering step, rank the default attributes/measures at top under their containing dimensions/facts (line 7-9). The algorithm counts the number of dimension elements and the number of fact elements. It places type of elements at top whose count is greater (line 10). Further, non-conformed dimension(s) filter the fact elements (line 11-13). If two keywords are part of the same element then combine them and rank the element at top (line 14-16).

5. Searching Data Instances

The purpose of the identification of elements before searching in the data instances is to reduce search space in terms of number of dimensions to be searched, number of attributes and at granularity the number of values. In this way, data instances do not have to be explored blindly for a keyword rather guided search is performed.

- Searching Identified Elements: The value keywords specified in the natural language query are searched within the identified attribute.
- Searching in Proximal Elements: The value keywords may not be mapped in the identified attributes. Those are then searched in the adjacent attributes of the matches.
- Level-wise Search: In case of failure of the retrieval of value keyword in identified and proximal attributes, perform a level-wise search. Initially search in default attributes of the atomic dimensions then in their adjacent attributes. If those are not found in atomic dimensions, extend search in next level of dimensions with similar procedure adopted for the atomic dimensions.

6. Experimental Results

The experiments have been performed on core i3 system with 2GB RAM. The LSM technique is evaluated by completeness, accuracy and performance parameters.

6.1. Dataset

The dataset includes a data warehouse and a query set taken for the solution evaluation.

- Data Warehouse: According to best of our knowledge, no bench mark is available yet for natural language interfaces to data warehouse. A parallel system KDAP [24] uses a data warehouse AdventureWorksDW which is the sample data warehouse provided by the ms sql server. Thus, we also use same data warehouse for experimental analysis and for comparison with parallel systems. The detail of the data warehouse schema is given in Table 6.

Table 6. Dataset specification.

Elements	Count
Dimensions	16
Fact Tables	6
Dimension Text Attributes	108
Dimension Non-Text Attributes	51
Dimension Tuples	22, 206
Total Tuples	2,63,869

- Query Set: A set of 50 natural language queries is taken for experimental analysis. Half of the queries are shown in Table 7. Queries are built by two type of users i.e., novice, and expert.

Table 7. Natural language query set.

1.	Show profit of Germany in US dollar currency for year 2000.
2.	Display sales of mountain tire in month of November and December.
3.	Mountain Tire Product.
4.	Give highest order quantity made from North west region.
5.	Average Discount amount in US dollar on Mountain End Caps purchase.
6.	Metal Tread Plate.
7.	Sales ratio of Ian Jenkins in 1st quarter 2013.
8.	Profit in 2002.
9.	Number of Seat tube sale in 2002 in Australia.
10.	HL Shell product yearly sale in North East.
11.	Average leave hours of employees in 2004.
12.	Total employees having marital status unmarried.
13.	Show product that has highest cost.
14.	South west value added reseller.
15.	Volume discount for warehouse reseller.
16.	Half price pedal sale in 2003.
17.	Show bearing ball sale handled by David in July.
18.	Fork end sale by employee Steven in Australia.
19.	Discount on HL Grip Tape.
20.	Total tax amount on Guide Pulley.
21.	Quantity purchased in Canada in 2nd quarter.
22.	Quantity ordered in United Kingdom in 2004.
23.	Tax amount generated in January 2002.
24.	Return Sales in 2001 and 2002.
25.	List unit price of all the products sold in year 2001.

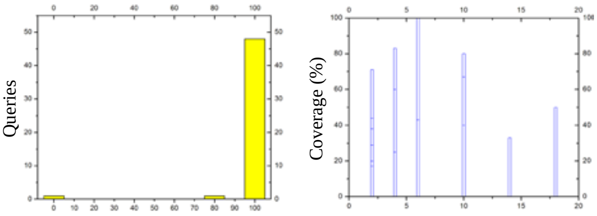
6.2. Comparative Analysis

In this section, LSM technique is being evaluated in comparison with existing solutions. None of the available systems perform keyword processing on logical schema level. Only Q and A system maintains a domain thesaurus generated from the logical schema and dimension values. The Q and A retrieves matches but does not rank them. Existing systems directly explore the data instances therefore comparison has been performed with LSM technique at data instances level. The comparison statistics is given in Table 8.

Table 8. Comparison statistics.

Q#	# Keywords	Non-Value Keywords	Ranking Accuracy	# Mapped Keywords	Coverage	# Dims.	# Rows	# Cols.	Values		
									LSM	KDAP	Q and A and QueGen
1	7	3	100%	3	43%	8	21377	8	85508	2843869	3809008
2	6	2	100%	2	33%	8	21377	14	85508	2843869	3809008
3	3	1	100%	1	33%	1	606	1	1212	1218801	1632432
4	6	4	100%	4	67%	9	666	5	1332	2437602	3264864
5	9	3	75%	4	44	8	21377	8	128262	3656403	4897296
6	3	0	0%	0	0	0	0	0	0	1218801	1632432
7	7	2	100%	2	29	6	20380	6	142660	2843869	3809008
8	2	1	100%	1	50	8	21377	8	42754	812534	1088288
9	6	2	100%	2	33	8	21377	8	85508	2437602	3264864
10	7	3	100%	3	43	8	21377	8	85508	2843869	3809008
11	5	4	100%	4	80	7	2893	7	2893	2031335	2721720
12	5	4	100%	4	80	1	296	1	296	2031335	2721720
13	3	3	66%	3	100	1	606	1	0	1218801	1632432
14	5	1	50%	2	40	7	2893	7	11572	2031335	2721720
15	4	2	100%	2	50	8	21377	8	42754	1625068	2177296
16	5	1	50%	2	40	8	21377	8	85508	2031335	2721720
17	5	1	100%	1	20	7	2893	7	11572	2031335	2721720
18	6	2	100%	2	33	7	2893	7	11572	2437602	3264864
19	4	1	100%	1	25	8	21377	8	64131	1625068	2177296
20	5	3	100%	3	60	8	21377	8	42754	2031335	2721720
21	4	2	100%	2	50	8	21377	8	42754	1625068	2177296
22	5	2	100%	2	40	8	21377	8	64131	2031335	2721720
23	4	2	100%	2	50	8	21377	8	42754	1625068	2177296
24	3	1	100%	1	33	8	21377	8	42754	1218801	1632432
25	6	5	100%	5	83	1	1158	1	1158	2437602	3264864

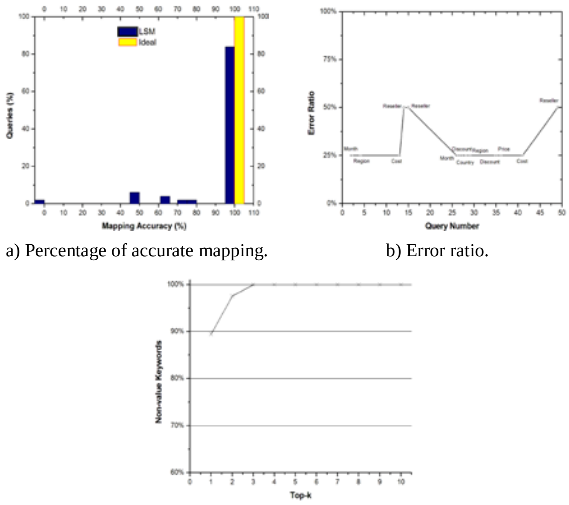
- Completeness: The schema mapping returns matches for non-value keywords whereas value keywords are mapped in the data instances according to the values searching mechanism discussed in section 5. Based on query set analysis provided in Table 8, non-value keywords are approximately 50% of the total keywords in average. Figure 2-a demonstrates that non-value keywords of 48 queries out of 50 have been 100% mapped in the logical schema. Only 1 query was 0% mapped because query#2 contained 0 non-value keywords. The query coverage is presented in Figure 2-b. It is calculated as # mapped keywords/ total keywords of query * 100 that gives percentage of query coverage. The average query coverage is 52.4%. There is a limitation of the system: If query coverage is reduced, scope of the schema mapping is also lessened.



a) Non-value keywords mapping. b) Query coverage in logical schema.

Figure 2. Completeness of mapping in logical schema.

- Accuracy: The accuracy of the proposed algorithm has been evaluated and shown in Figure 3. As there is no competitor system at schema mapping level, so accuracy is tested in comparison to an ideal case. As depicted in Figure 3-a, 84% of the queries have gained 100% accuracy. Few keywords from remaining 16% queries could not be accurately ranked. For such keywords, error ratio has been calculated. The error ratio is computed as: If the desired element is available in matches but could not be ranked at top, it is weighted 25%, 50% if found in atomic dimensions, 75% if retrieved from 2nd level dimensions and 100% for 3rd level dimensions.



a) Percentage of accurate mapping.

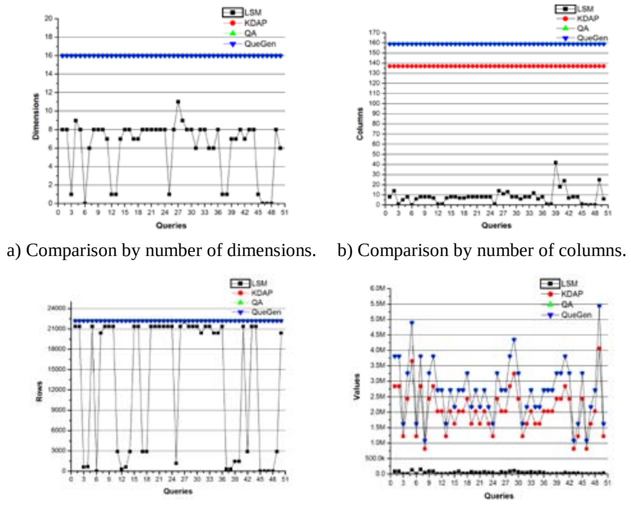
b) Error ratio.

c) Evaluation of the ranking.

Figure 3. Accuracy test.

For instance, keyword ‘month’ in query 2 is available in matches list thus has 25% error ratio. This phenomenon is illustrated in Figure 3-b. In comparison to other systems, even a keyword having 100% error ratio returns result faster by using our technique. Figure 3-c shows the accuracy by top-k ranked matches. The x-axis represents the top-k ranked results and the y-axis corresponds to the percentage of the non-value keywords satisfied. A point (x, y) on the curve, represents that the most relevant matches of y percent of the keywords, can be found in the top-k matches. As we can see, our LSM technique rank 89.43% of the keyword matches in the top 1, and overall, our LSM technique managed to reveal the relevant matches in the Top 3. The LSM technique is constrained while rank numeric values because same value may exist in multiple columns.

- Performance: The performance of our LSM technique is evaluated in comparison to the existing systems. It is checked by the number of dimensions, number of rows, number of columns and number of values explored. Figure 4-a depicts the searching of the data instances by number of dimensions. All parallel systems including KDAP, Q and A and QueGen explore all 16 dimensions for each query whereas, LSM technique scans minimum dimensions such as it explores maximum 11 dimensions in query 27. Further, comparison is presented by number of dimensional columns in Figure 4-b. The Q and A and QueGen search 159 columns whereas KDAP examine 137 textual columns. Our system outperforms the other systems as it explores maximum 42 columns in query 39. Figure 4-c shows comparison based on number of rows. Other systems scan all 22, 206 dimension rows. Our system explores fraction of rows as query 27 explore maximum 22, 083 rows. Finally, Figure 4-d gives comparison along number of values. It can be seen, Q and A and QueGen examine maximum 54, 41,440 values. As KDAP only search in textual columns, it explores maximum 40, 62, 670 values in the data instances. Our system searches maximum 1, 42,660 values. Therefore, LSM technique outperforms at data instances level.



a) Comparison by number of dimensions. b) Comparison by number of columns.

c) Comparison by number of rows. d) Comparison by number of values.

Figure 4. Performance test.

7. Conclusions and Future Work

In this paper, a technique to explore data efficiently in the data warehouse is proposed for the natural language interface. The natural language query input of the user is mapped to the logical schema to identify the elements. As a result, a number of matches for each keyword are acquired which are ranked according to the six criteria. An algorithm has been implemented based on the criteria. The value keywords are explored inside ranked elements in the data instances. If value could not be found in the ranked elements, proximal and level-wise search is performed.

The proposed LSM technique has been evaluated on the AdventureWorksDW data warehouse. It is tested for completeness, accuracy and performance parameters. The test conducted to evaluate the completeness, results shows that non-value keywords of 88% queries were 100% mapped in the schema while approximately 50% of the query keywords have coverage in the schema.

In continuation of accuracy test, 84% of the queries from query set were accurately mapped. Further, our technique rank 89.43% of the keyword matches in the top 1, and overall, LSM technique managed to reveal the relevant matches in the top 3. Finally, experiments have been carried out for performance evaluation according to which our system outperforms the existing systems.

As future work, investigation is required to extend the LSM technique to efficiently build the aggregations. The rollup, drill-down, slice and dice operations should be manipulated with maximum ease for non-technical users. Work is required to improve query coverage in case of minimum non-value keywords present in user query. The investigation is needed to accurately rank the numeric values existing in user query and which establish selection constraints.

References

[1] Agrawal S., Chaudhuri S., and Das G., “Dbxplorer: A System for Keyword-Based Search Over Relational Databases,” *in Proceeding the 18th International Conference on Data Engineering*, DC, pp. 5-16, 2002.

[2] Aljanabi A., Alhamami A., and Alhadidi B., “Query Dispatching Tool Supporting Fast Access to Data Warehouse,” *The International Arab Journal of Information Technology*, vol. 10, no. 3, pp. 269-275, 2013.

[3] Androutsopoulos I., Ritchie G., and Thanisch P., “Natural Language Interfaces to Databases-An Introduction,” *Natural Language Engineering*, vol. 1, pp. 29-81, 1995.

[4] Bao Z., Ling T., Chen B., and Lu J., “Effective XML Keyword Search with Relevance Oriented Ranking,” *in Proceeding of IEEE International Conference on Data Engineering*, Shanghai, pp. 517-528, 2009.

[5] Bhalotia G., Hulgeri A., Nakhe C., Sudarshan S., and Chakrabarti S., “Keyword Searching and Browsing in Databases using BANKS,” *in Proceeding of 18th International Conference on Data Engineering*, CA, pp. 431-440, 2002.

[6] Bruckner R. and Tjoa A., “Managing Time Consistency for Active Data Warehouse

Environments," *in Proceeding of 3rd International Conference on Data Warehousing and Knowledge discovery*, London, pp. 254-263, 2001.

[7] Chaudhuri S. and Dayal U., "An Overview of Data Warehousing and OLAP Technology," *SIGMOD Record*, vol. 26, no. 1, pp. 65-74, 1997.

[8] Chaudhuri S., Ramakrishnan R., and Weikum G., "Integrating DB and IR Technologies: What is the Sound of One Hand Clapping?," *in Proceeding of the Conference on Innovative Data Systems Research (CIDR)*, CA, 2005.

[9] El-Mouadib F., Zubi Z., Almagrous A., and El-Feghi I., "Generic Interactive Natural Language Interface to Databases (GINLIDB)," *International Journal of Computers*, vol. 3, no. 3, 2009.

[10] Golenberg K., Kimelfeld B., and Sagiv Y., "Keyword Proximity Search in Complex Data Graphs," *in Proceeding of 2008 ACM SIGMOD International Conference on Management of Data*, Canada, pp. 927-940, 2008.

[11] Guo L., Shao F., Botev C., and Shanmugasundaram J., "XRANK: Ranked Keyword Search over XML Documents," *in Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, California, pp. 16-27, 2003.

[12] Hristidis V., Gravano L., and Papakonstantinou Y., "Efficient IR-Style Keyword Search over Relational Databases," *in Proceedings of the 29th International Conference on Very Large Data Bases*, Berlin, pp. 850-861, 2003.

[13] Hristidis V. and Papakonstantinou Y., "Discover: Keyword Search in Relational Databases," *in Proceeding of the 28th International Conference on Very Large Data Bases*, Hong Kong, pp. 670-681, 2002.

[14] Hristidis V., Papakonstantinou Y., and Balmin A., "Keyword Proximity Search on XML Graphs," *in Proceeding of 19th International Conference on Data Engineering*, Bangalore, pp. 367-378, 2003.

[15] Huang B., Zhang G., and Sheu P., "A Natural Language Database Interface Based on a Probabilistic Context Free Grammar," *in Proceeding of IEEE International Workshop on Semantic Computing and Systems*, Huangshan, pp. 155-162, 2008.

[16] Kabra N., Ramakrishnan R., and Ercegovac V., "The QUIQ Engine: A Hybrid IR DB System," *in Proceeding of 19th International Conference on Data Engineering*, Bangalore, pp. 741-743, 2003.

[17] Kacholia V., Pandit S., Chakrabarti S., Sudarshan S., Desai R., and Karambelkar H., "Bidirectional Expansion for Keyword Search on Graph Databases," *in Proceeding of the 31st Very Large Data Bases Conference*, Norway, pp. 505-516, 2005.

[18] Kuchmann-Beauger N. and Aufaure M., "A Natural Language Interface for Data Warehouse Question Answering," *in Proceeding of 16th International Conference on Applications of Natural Language to Information Systems*, Alicante, pp. 201-208, 2011.

[19] Liu F., Yu C., Meng W., and Chowdhury A., "Effective Keyword Search in Relational Databases," *in Proceeding of the 2006 ACM SIGMOD International Conference on Management of Data*, Chicago, pp. 563-574, 2006.

[20] Majeed F. and Shoaib M., "A Natural Language Based Retrieval System for the Data Warehouse," *Pakistan Journal of Science (PJS)*, vol. 65, no. 3, pp. 426-434, 2013.

[21] Naeem M., Saif U., and Bajwa I., "Interacting with Data Warehouse by Using a Natural Language Interface," *in Proceeding of 17th International Conference on Applications of Natural Language to Information Systems*, Netherlands, pp. 372-377, 2012.

[22] Sapia C., Blaschka M., Höfling G., and Dinter B., "Extending the E/R Model for the Multidimensional Paradigm," *in Proceeding of the Workshops on Data Warehousing and Data Mining: Advances in Database Technologies*, London, pp.105-116, 1998.

[23] Stratica N., Kosseim L., and Desai B., "Using Semantic Templates for a Natural Language Interface to the CINDI Virtual Library," *Data and Knowledge Engineering*, vol. 55, no. 1, pp. 4-19, 2005.

[24] Wu P., Sismanis Y., and Reinwald B., "Towards Keyword-driven Analytical Processing," *in Proceeding of the 2007 ACM SIGMOD International Conference on Management of Data*, Beijing, pp. 617-628, 2007.

[25] Yu j., Qin L., and Chang L. "Keyword Search in Relational Databases: A Survey," *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 2010.

Fiaz Majeed is working as Lecturer under faculty of Computing and Information Technology at University of Gujrat (UOG), Pakistan. He received MS in Computer Science from COMSATS Institute of Information Technology (CIIT) Lahore Pakistan in 2009. He is currently PhD scholar in University of Engineering and Technology Lahore Pakistan. His research interests include data warehousing, Natural Language Processing and information retrieval. He has published 15 research papers in refereed journals and international conference proceedings in the above areas. He is doing his Ph.D. under the supervision of Prof. Dr. Muhammad Shoaib.

Muhammad Shoaib is Professor at Computer Science and Engineering Department at the University of Engineering and Technology (UET) Lahore, Pakistan. He completed his Ph.D. from the University of Engineering and Technology, Lahore, Pakistan in 2006. His Post Doc. is from Florida Atlantic University, USA, in 2009. His current research interests include Information Retrieval (IR) Systems, Information Systems, Software Engineering and Semantic Web. He has published more than 40 papers in refereed journals and international conference proceedings in the above areas.