IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

Database Queries Using Dynamic Query Forms

Priyanka G. Varkad[1], Prof. P. N. Kalavadekar[2]

PG Student, Computer Engineering, SRES's College of Engineering, Kopargaon, India [1]

Assistant Professor, Computer Engineering, SRES's College of Engineering, Kopargaon, India[2]

Abstract: With the rapid enhancement of web services and their popularity, web users are increasing day by day. The modern databases are relational and include large number of relations and attributes. By this enhancement of web information and scientific databases it is not able to get user require results with the static query forms. The solution to this problem is dynamic queries. This paper provides a Dynamic Query Form (DQF), a curious database query form interface which is able to tackle the large and complex relational databases. A system captures the user's preference during the user communication and guides user to make decisions. Query form generation is a iterative process. The ranking of form components is based on the captured user preferences. A user can also fill up the query form and deliver queries to view the query output at each step. In this way, a query form could be dynamically refined till the user gets the query result.

Keywords: Query Form, User Interaction, Query Form Generation, Query Form Enrichment, and Dynamic Query forms (DQF).

I. INTRODUCTION

'Query Form' is one of the most extensively used user interface for querying databases to access information. The traditional query forms are configured and predefined by the Developers and Database Administrators in various information management systems. With the rapid development of web informatics and scientific databases, the modern databases have become very huge and complex. There are over hundreds of entities for biological and chemical data resources in the databases in natural sciences, such as diseases and genomics. The web databases, like DBPedia and Freebase, usually have over thousands of structured web entities. So, it is difficult to design a set of static query forms which better response the different ad-hoc database queries on those complex databases. Many modern database management and development tools, such as SAP and MS Access, allows user to develop customized queries on databases, by providing several mechanisms. The development of these customized queries totally based upon manual editing's of user. If user is not familiar with the database schema in advance, he/she will be confused by the hundreds and thousands of data attributes. Tackling with the relational database is a challenging task for non-technical user. Considering this view, in recent years many researchers are focusing on database interfaces so that user can query the relational databases with no SQL easily.

This paper proposes a Dynamic Query Form (DQF) system, an interface which is capable of generating query forms for user at runtime. Different from traditional document retrieval, prior to identify the final candidate, the users in database retrieval need to execute several rounds of action [6]. The important features of DQF includes: a) Capture the user interest during the user interaction and b) Iteratively adapt the query forms.

The rest of the paper is arranged as follows: Section 2 presents the literature survey over the related work. Section 3 gives in brief idea about the existing system and proposed system. Finally, the section 4 concludes the review paper.

II. LITERATURE SURVEY

Researchers focus is on database interfaces which assist users to query the relational database with no SQL. There are two most widely used database querying interfaces: QBE (Query by Example) and Query Form. Current studies and works mainly focus on how to create the query forms.

A. Automated creation of a forms based database query interface:
Jayapandian and H. V. Jagadish, in their paper stated that various existing database management and development tools, such as EasyQuery, Cold Fusion, SAP and Microsoft Access, provide several mechanisms to let users create customized queries on databases. However, the creation of customized queries totally depends on user's manual editing. If a user is not familiar with the database schema in advance, those hundreds or thousands of data attributes would confuse him/her. It first finds a set of data attributes, which are most likely queried based on the database schema and data instances. After that, the query forms are generated based on the chosen attributes [1].

B. Automating the design and construction of query forms:
M. Jayapandian and H. V. Jagadish, proposed a workload driven method. It aims to find the representative queries by applying the clustering algorithm. Based on those

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

representative queries the query forms are generated. One problem of the aforementioned approaches is that, in case of lots of query forms generation in advance, there are still user queries which cannot be satisfactorily solved by any one of query forms. Another problem is that, when we create a huge number of query forms, it's a difficult task to let users find an appropriate query form would be difficult. The Query interfaces play a vital role in determining the usefulness of a database. A form-based interface is widely regarded as the most user-friendly querying method. In this paper, they developed mechanisms to defeat the challenges that limit the usefulness of forms, namely their restrictive nature and the tedious manual effort required to build them. Specifically, they introduce an algorithm to generate a set of forms automatically given the expected query workload [2].

C. Combining keyword search and forms for ad hoc querying of databases:
E. Chu, A. Baid, X. Chai, A. Doan, and J. F. Naughton, provides solution that combines keyword search with query form generation. The solution is, in advance to generate a lot of query forms automatically. User can find relevant query forms from a large number of pre-generated query forms by giving it several keywords as an input. The user inputs several keywords to find related query forms from a huge number of previously generated query forms but it is not suitable when the user does not have real keywords to describe the queries [3].

D. Automated ranking of database query results:
S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis, state that the results of a query is a popular aspect of the query model in Information Retrieval (IR) that we have grown to depend on. In contrast, database systems support only a Boolean query model. For instance, a selection query on a SQL database returns all tuples that satisfy the conditions in the query. Hence, the following two situations are not gracefully handled by a SQL system: Empty answer and many answer [4].

E. Query recommendations for interactive database exploration:
G. Chatzopoulou, M. Eirinaki, and N. Polyzotis, stated that now day there are numerous ways to explore the database in order to recommend the query forms. SQL queries play a vital role to recommend the user related queries as per their intendment. However they are not considering the quality of query forms much. Here is an additional method to recommend based on query results. The differences between these two strategies are each and every loop will provide the query component but in the other hand of previous recommendation is providing complete query [5].

F. Facetedpedia: Dynamic generation of query-dependent faceted interfaces for Wikipedia:
C. Li, N. Yan, S. B. Roy, L. Lisham, and G. Das, Dynamic faceted search is a type of search engines where relevant facets are presented for the users according to their navigation paths. Dynamic faceted search engines are similar to our dynamic query forms if we only consider Selection components in a query. However, besides Selections, a database query form has other important components, such as Projection components. Projection components control the output of the query form and cannot be ignored. Moreover, designs of Selection and Projection have inherent influences to each other [6].

G. Dynamic Query Forms for Database Queries:
L. Tang, T. Li, Y. Jiang, Z. Chen, provides a solution that nontechnical users make usage of relational database which is a challenging task. Therefore, in modern years lots of researches were focused on database interfaces to help users to query the relational databases without use of SQL. This paper provides a Dynamic Query Form system (DQF), is a query interface able to dynamically produce query forms for the users. Unlike conventional document retrieval, before distinguishing the final candidate, the users in database retrieval are mostly willing to execute several rounds of action [7].

III.SYSTEM ARCHITECTURE

A. Proposed System
The proposed system has breakdown structure as follow shown in fig. 1.

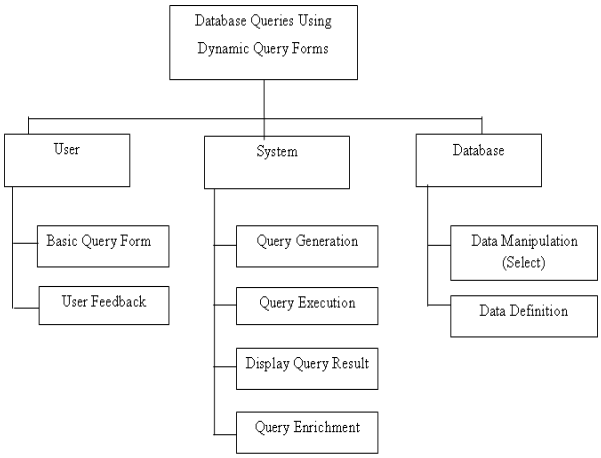


Fig 1: The breakdown Structure.

The breakdown structure is shown in above fig. which mainly focuses on following modules:

B. User
The user interacts with the system to access the database. This module has two sub modules.

- Basic Query Forms :

The user fills the present query form and submits a query. The basic query form includes very few primary attributes of the database. The basic query form is then enriched iteratively by exchanging between the user and system until the user is satisfied with the query result. Each query

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

form corresponds to an SQL query pattern. A query form F is defined as a tuple (A_F, R_F, σ_F, $\bowtie(R_F)$), which represents database query pattern as follows:

F = (SELECT A_1, A_2,……., A_K
FROM $\bowtie$ (R_F) WHERE σ_F),
Where, A_F = {A_1, A_2,……,A_k} are k attributes for projection, k > 0. R_F = {R_1, R_2,……,Rn} is the set of n relations (or entities) concerned in this query, n > 0. Each attribute in A_F belongs to one relation in R_F. σ_F is a conjunction of expressions for selections on relations in R_F. $\bowtie$ (R_F) is a join function to make a conjunction of expressions for joining relations of R_F. In the user interface of a query form F, A_F is the set of columns of the result table. σ_F is the set of input components for users to fill. Query forms allow users to fill parameters to generate different queries. R_F and $\bowtie$ (R_F) are not visible in the user interface, which are usually generated by the system according to the database schema. For a query form F, $\bowtie$ (R_F) is automatically constructed according to the foreign keys among relations in R_F. Meanwhile, R_F is determined by A_F and F. R_F is the union set of relations which contains at least one attribute of A_F or σ_F. Hence, the components of query form F are actually determined by A_F and σ_F. As we mentioned, only A_F and σ_F are visible to the user in the user interface.

• User Feedback:
This module is intended for taking user feedback on the basis of result shown for query execution. The user feedback will be in the form of whether user is satisfied or not. This will help the user to alter the query and get best desired results.

C. System:
Fig 2, begins with a basic query form which contains very few primary attributes of the database. The user's fills out a current query form and submit it. Query is executed by the system and it also display a query result at a user side, and if user is not satisfied with the result then basic query form is enriched iteratively between the user and our system until the user is fulfilled with the query results.

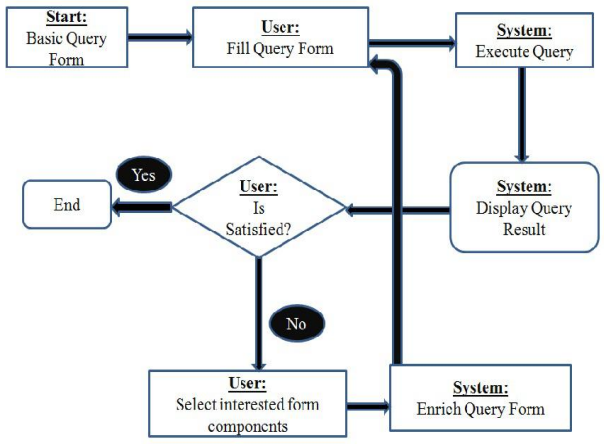


Fig 2: Flowchart of Dynamic Query Form.

• Query Generation:
This module will provide an easy way to handle GUI for end users to generate the SQL queries. This will help end users to retrieve data from the database without the knowledge of the SQL. The GUI will contain different components such as List of tables, provision for applying different conditions, provision for selecting one or more column names etc. The query construction algorithm is used for generating a query which is useful to display the data at user side. Query is generated by using query construction algorithm which is as follow:

Query Construction for Select Query.
1. Select attributes from tablename
2. String s = "select"
3. String[] column
4. For(i to column. length; i++)
5. column [i] = column
6. String column = join(column, " ")
7. String tablename
8. String query = s + column + tablename

• Display Query Result:
System displays a query result at user side.
• Query Execution:
Initially the user fills out the current query form and submits a query. After that DQF executes the query and display the results at a user side.

TABLE I INTERACTIONS BETWEEN USERS AND DQF

1	Query Form Enrichment	1) The user fills out the current query form and submit a query. 2) DQF executes the query and shows the results. 3) The user provides the feedback about the query results.
2	Query Form Enrichment	1) DQF recommends a ranked list of query form components to the user. 2) The user selects the desired form components into the current query form.

• Query Enrichment:
This module will have previously executed query and it will allow the user to alter it for getting best results as per his/her needs. The query enrichment is the process in which query gets altered and developed on the basis of user feedback. This will be the repetitive process till user gets the outcomes as per his/her needs and gets satisfied. And DQF also recommends a ranked list of query form components to the user.

D. Database:
A database is an accumulation of information that is organized so that it can be straightforwardly accessed, managed, and updated. A database module consists of two sub-modules i.e. Data Manipulation and Data Definition.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

A data manipulation language (DML) is a group of computer languages including commands that permit users to manipulate data in a database. This manipulation includes inserting data into database tables, recovering existing data, deleting data from existing tables and modifying existing data. The DML operation are SELECT, INSERT, UPDATE, DELETE.

Data definition Language (DDL) is a standard for commands that describe the diverse structures in a database. DDL statements create, modify, and remove database objects such as tables, indexes, and users. Regular DDL statements are CREATE, ALTER, DROP.

IV. EXPERIMENTAL RESULTS

The proposed system presented in the paper works in a modular approach thereby making the system work in a sequential manner. The following snapshots show how system works and display the data to the user without writing the SQL query. The outputs of the implemented modules of the proposed system are as follows:

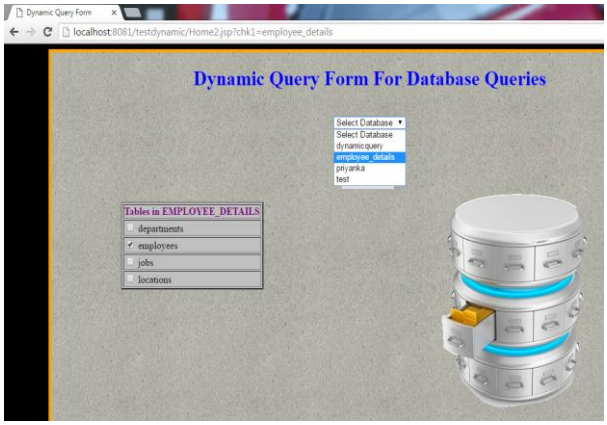


Fig 3: Selecting database and table name.

Fig 3. Show the database is selected firstly and database display the tables present in it. User than select the table from the database.

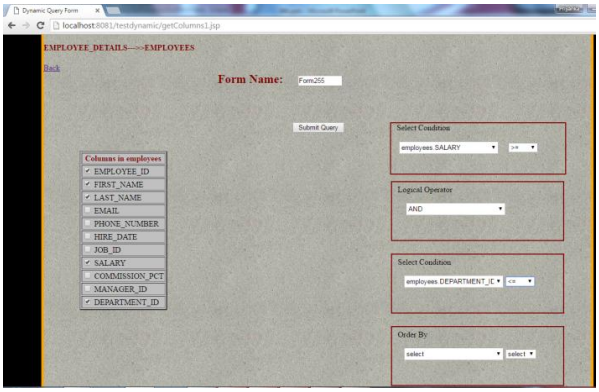


Fig 4: Selecting Condition.

Fig 4. Shows the new form is generated and user fills the query form. Firstly user selects the attributes from the table and then selects the condition and operator. User is also free to select two conditions and use a logical operator.

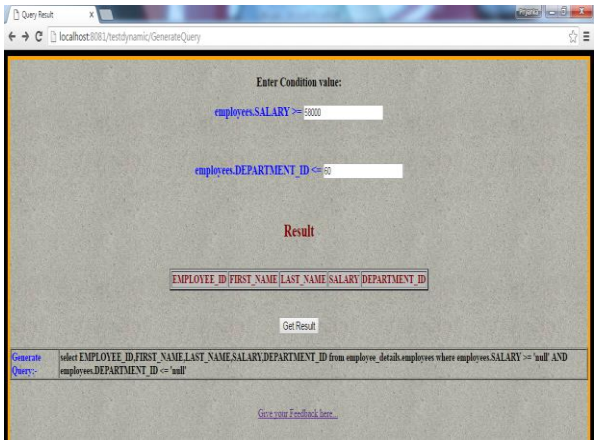


Fig 5: Query Generation and Execution.

Fig 5, Shows a query generation and execution User enter a value in fields and based on the values query is generated and fired on database.

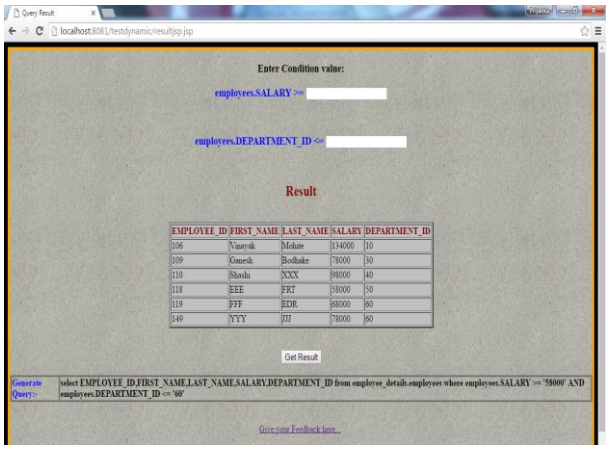


Fig 6: Display Result.

Fig 6, Shows that the result is displayed on the user's side and it also shows a SQL query which is generated by selecting a attributes from the table.

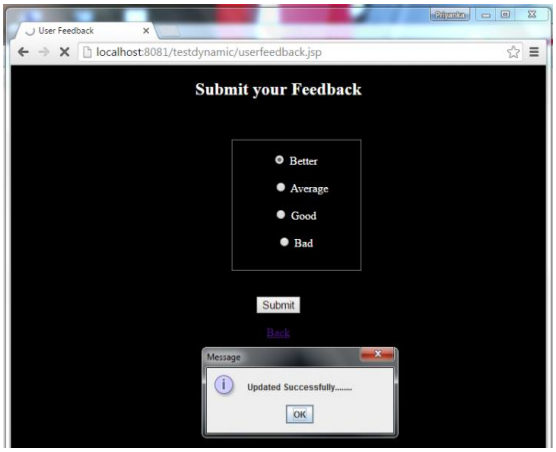


Fig 7: User Feedback.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

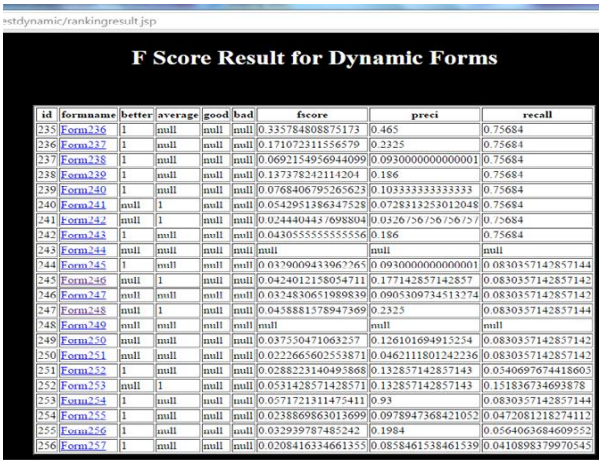
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

A. Experimental Setup:

The system has implemented as standalone system using java-5.1.12. The system uses SQL Server as the database engine. All experiments are run using machine with Intel(R) Core (1M) i3-4005U CPU @ 1.70GHz 1.70GHz, and running on Windows XP. Employee data-set is used for this experiment. Employee dataset consists of four databases, 15 tables, 25 attributes, and near about 500 records from all the tables.

B. Result Analysis:

In result analysis the F-score is calculated with the help of precision and recall. F-score is used to measure the goodness of query forms. Also the feedbacks backup is store so that we come to know what feedback is given by the user to a particular form. Also by clicking on the form name shown in the fig 8 shows that one can edit form again and also view the result whenever needed.



id	formname	better	average	good	bad	fscore	preci	recall
235	Form236	1	null	null	null	0.3357848088751 73	0.465	0.75684
236	Form237	1	null	null	null	0.171072311556579	0.2325	0.75684
237	Form238	1	null	null	null	0.0692154956944099	0.0930000000000001	0.75684
238	Form239	1	null	null	null	0.137378242114204	0.186	0.75684
239	Form240	1	null	null	null	0.0768406795265623	0.103333333333333	0.75684
240	Form241	null	1	null	null	0.0542951386347528	0.0728313253012048	0.75684
241	Form242	null	1	null	null	0.0244404437698804	0.0326756756756757	0.75684
242	Form243	1	null	null	null	0.0430555555555556	0.186	0.75684
243	Form244	null	null	null	null	null	null	null
244	Form245	1	null	null	null	0.0379009433962265	0.0930000000000001	0.0830357142857144
245	Form246	null	1	null	null	0.0424012158054711	0.177142857142857	0.0830357142857142
246	Form247	null	null	null	null	0.0324830651989839	0.0905309734513274	0.0830357142857142
247	Form248	null	1	null	null	0.0458881578947369	0.2325	0.0830357142857144
248	Form249	null	null	null	null	null	null	null
249	Form250	null	null	null	null	0.037550471063257	0.126101694915254	0.0830357142857142
250	Form251	null	null	null	null	0.0222665602553871	0.0462111801242236	0.0830357142857142
251	Form252	1	null	null	null	0.0288223140493868	0.132857142857143	0.0540697674418603
252	Form253	null	1	null	null	0.0531428571428571	0.132857142857143	0.151836734693878
253	Form254	1	null	null	null	0.0571721311475411	0.93	0.0830357142857144
254	Form255	1	null	null	null	0.0238869863013699	0.0978947368421052	0.0472081218274112
255	Form256	1	null	null	null	0.032939787485242	0.1984	0.0564063684609552
256	Form257	1	null	null	null	0.0208416334661355	0.0858461538461539	0.0410898379970545

Fig 8: F-Score Result.

V. CONCLUSION

This paper state that a DQF i.e., Dynamic Query Form generation approach which help users dynamically generate query forms. The system is intended to generate different types of queries on the basis of user feedback with iterations of the query enrichment process. A user can also fill the query form and submit queries to analysis the query result at each iteration. The dynamic query form generation system focuses on providing user friendly GUI, efficient and fast query generation, and query enrichment.

ACKNOWLEDGMENT

I would like to take this opportunity to express my heartfelt thanks to my guide **Prof. P. N. Kalavadekar** for his esteemed guidance and encouragement, especially through difficult times. His suggestions broaden my vision and guided me to succeed in this work. I am also very grateful for his guidance and comments while designing part of my research paper and learnt many things under his leadership.

REFERENCES

[1] M. Jayapandian and H. V. Jagadish, "Automated creation of a forms-based database query interface," Proc. VLDB, vol. 1, no. 1, pp. 695709, Aug. 2008.

[2] M. Jayapandian and H. V. Jagadish, "Automating the design and Construction of query forms," IEEE Trans. Knowl. Data Eng., vol. 21, no. 10, pp. 13891402, Oct. 2009.

[3] E. Chu, A. Baid, X. Chai, A. Doan, and J. F. Naughton, "Combining keyword search and forms for ad hoc querying of databases," in Proc. ACM SIGMOD, Providence, RI, USA, Jun. 2009, pp. 349360.

[4] S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis, "Automated ranking of database query results," in Proc. CIDR, 2003 pp. 888899.

[5] G. Chatzopoulou, M. Eirinaki, and N. Polyzotis, "Query recommendations for interactive database exploration," in Proc. SSDBM, New Orleans, LA, USA, Jun. 2009, pp. 318.

[6] C. Li, N. Yan, S. B. Roy, L. Lisham, and G. Das, "Facetedpedia: Dynamic generation of query-dependent faceted interfaces for wikipedia," in Proc. WWW, Raleigh, NC, USA, Apr. 2010, pp. 651660.

[7] Liang Tang, Tao Li, Yexi Jiang, and Zhiyuan Chen, "Dynamic Query Forms for Database Queries", IEEE Trans. on Knowledge and Data Engg. Vol:PP No:9 sept. 2014.

BIOGRAPHY

Miss. Priyanka G. Varkad received the B.E. degree in Computer Engineering from Savitribai Phule University of Pune, Pune, Maharashtra, India in 2014. She is currently pursuing the M.E. degree in Computer Engineering in SRES's Sanjivani College of Engineering, Kopargaon, Savitribai Phule Pune University, Pune, Maharashtra, India. Her current research interests include Data Mining and Information Retrieval.