Data Mining Using Dynamic Query Form Techniques

Pranali Dhamankar[1], Manisha Moundekar[2], Harshal Bansod[3] , Purwa Yawalkar[4] ,Archana Nikose[5]

[1,2,3,4]UG Students, [5]Assistant Professor
[1,2,3,4,5]Dept. of Computer Science And Engineering
Priyadarshini Bhagwati College of Engineering, Nagpur, Maharashtra

---***---

Abstract - A Dynamic query form system which generates query form according to the user's desire at the run time. The system provides a solution for the query interface in large and complex databases. Different kinds of databases such as modern scientific database, web dataset. These are maintain large and heterogeneous data with large number of relations and attribute. So it is very difficult to design a set of static question forms to answer various ad-hoc database queries on these modern database. Thus there is a need of such system which generate Query forms dynamically according to the user's need at run time. The proposed system Dynamic Query Form i.e. DQF system going to provide a solution by the query interface in large and complex databases. The goodness of a query form is determined by the query results generated from the query form.

Key Words: Query Forms, User interaction, Query Form Generation, DQF, Databases, query forms

1.INTRODUCTION

Data mining is a process of extracting hidden predictive information from large database or extracting knowledge from large amount of data is called as data mining. Data mining sources database for hidden patterns, finding predictive information that experts may miss, as it goes beyond their expectations. Knowledge discovery in database (KDD) is the process of discovering useful knowledge from the collection of data. The process of KDD are Data cleaning, Data integration, Data Selection, Data Transformation, Data mining, Pattern evaluation and Knowledge representation. Query is one of the most widely used user interfaces for querying databases. Traditional query forms are designed or predefined by developers. There are different types of query like Select, Insert, Update, Delete etc. Select is one of the most important form of query. In natural Sciences such as genomics and diseases, the database have over hundreds of entities for chemical and biological data resources. Many web databases, such as freebase and DB Pedia, typically have thousands of structured web entities. Therefore it is difficult to design a set of static query forms to satisfy various ad-hoc database queries on those complex database. Ranking and Indexing Technique is supported in this process of Dynamic query form for Database query. Ranking is a list of all items in a prominent place and Indexing is a way to optimize the performance of a database by minimizing the number of disk accesses required when query is processed. It is a data structure technique which is used to quickly locate and access the data in a database.

2. LITERATURE SURVEY

Eric Chu, Aakansha baid, Xiaoyong Chai, AnHain Doan, Jeffery Naughton [1] Specifically, we propose to take as input a target database and then generate and index a set of query forms offline. At query time, a user with a question to be answered issues standard keyword search queries; but instead of returning tuples, the system returns forms relevant to the question. The user may then build a structured query with one of these forms and submit it back to the system for evaluation.

Magesh Jayapandian , H.V.Jagadish [2] In this paper, we seek to maximize the ability of a forms-based interface to support queries a user may ask, while bounding both the number of forms and the complexity of any one form. While a careful analysis of real or expected query workloads are useful in designing the interface, these query sets are often unavailable or hard to obtain prior to the database even being deployed. Hence generating a good set of forms just using the database itself is a challenging yet important problem.

Shyam Boriah, Varun Chandola, Vipin Kumar [3] In this paper we study the performance of a variety of similarity measures in the context of a specific data mining task outlier detection. Results on a variety of data sets show that while no one measure dominates others for all types
of problems,some measures are able to have consistently high performance.

3. PROPOSED SYSTEM

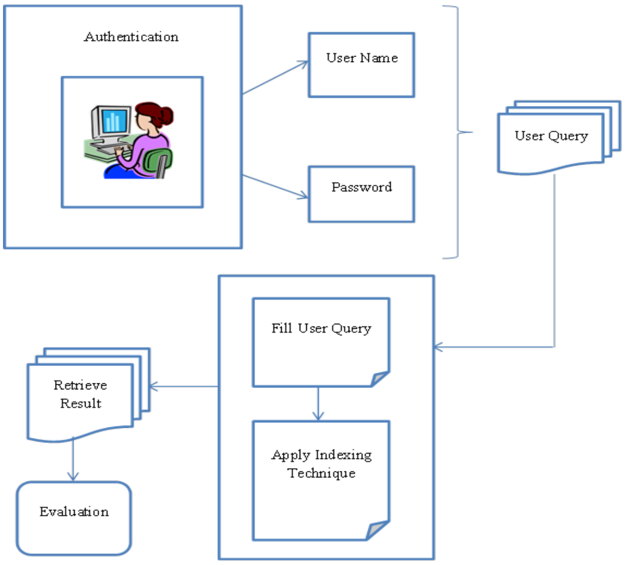
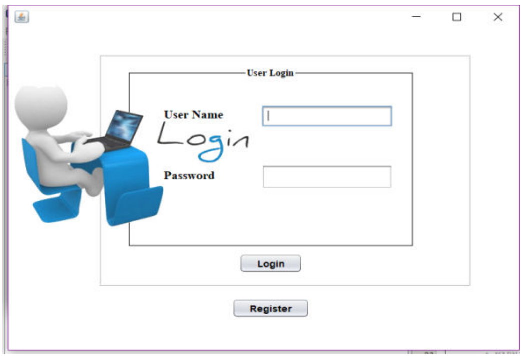


Fig (a) : Architecture of dynamic query form

With the quick improvement of web data and logical databases, current databases become extremely huge and complex. In common sciences, for example, genomics and maladies, the databases have more than several elements for compound and natural information assets. Many web databases, for example, Freebase and DB Pedia, commonly have a great many organized web elements. Accordingly, it is hard to structure a lot of static question structures to fulfill different impromptu database inquiries on those mind boggling databases. Many existing database the board and improvement devices, for example, Easy Query, Cold Fusion, SAP and Microsoft Access, give a few components to let clients make tweaked inquiries on databases. Many existing database the board and improvement devices, for example, Easy Query, Cold Fusion, SAP and Microsoft Access, give a few components to let clients make tweaked inquiries on databases. The essence of DQF is to capture user interests during user attraction and to adapt the query form iteratively until user is satisfied. Each iteration consists of two varieties of user interactions: query form enrichment and query execution. Fig (a) shows the work flow of dynamic query form. It starts with a basic query form which contains very few primary attributes of database. The basic query type is then enrich iteratively via the interaction between the user and system till the user is happy with the question result. Proposed system has following modules.

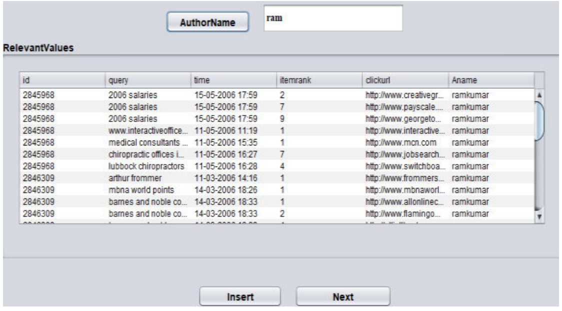
1. User Login
2. Query Execution
3. Database Query Recommendation
4. Indexing technique Retrieval
5. Performance Evaluation

1.User Login: The user through the credentials conferred by the user. If the user is new user login that is user entry level here the authentication that is completed supported the username and paasword. A login refers to the credentials needed to get access to a system or different restricted space. work in or on is that the method by that individual access to a system is controlled by distinctive and authenticating member he/she ought to register the login type to info question result and retrieve. This is the figure of user login page.



Fig(1): User login Page

2.Query Execution: The dataset that is gathered from UCI machine learning repository that is loaded and preprocess the information for improvement the input. By considering a general programee for specific period from that they need to gather info like Anonymous ID, query, query time, URL. From this dataset they need to extract needed info associated with our next step method. The user fills out this question kind and submits a question. DQF executes the question and shows the results. The user provides the feedback concerning the question results.



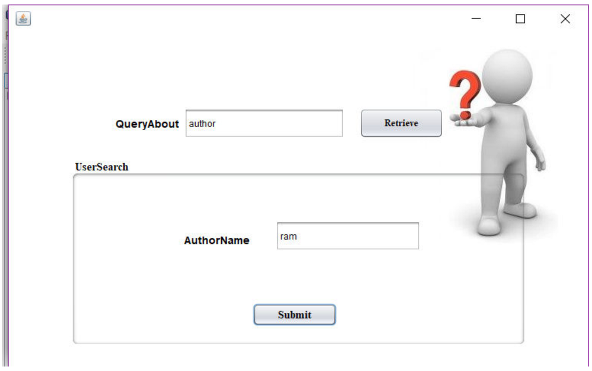
ram

RelevantValues

id	query	time	itemrank	clickurl	Aname
2845968	2006 salaries	15-05-2006 17:59	2	http://www.creativegr...	ramkumar
2845968	2006 salaries	15-05-2006 17:59	7	http://www.payscale....	ramkumar
2845968	2006 salaries	15-05-2006 17:59	9	http://www.georgeto...	ramkumar
2845968	www.interactiveoffice...	11-05-2006 11:19	1	http://www.interactive...	ramkumar
2845968	medical consultants ...	11-05-2006 15:35	1	http://www.mcn.com	ramkumar
2845968	chiropractic offices i...	11-05-2006 16:27	7	http://www.jobsearch...	ramkumar
2845968	lubbock chiropractors	11-05-2006 16:28	4	http://www.switchboa...	ramkumar
2846309	arthur frommer	11-03-2006 14:16	1	http://www.frommers...	ramkumar
2846309	mbna world points	14-03-2006 18:26	1	http://www.mbnaworl...	ramkumar
2846309	barnes and noble co...	14-03-2006 18:33	1	http://www.allonlinec...	ramkumar
2846309	barnes and noble co...	14-03-2006 18:33	2	http://www.flamingo...	ramkumar

Fig(2): Query Execution Page

3.Database Query Recommendation: Recent studies introduce collaborative approaches to recommend database query components for database exploration. To assist users in this context, we draw inspiration from Web recommender systems and propose the use of personalized query recommendations. The idea is to track the querying behavior of each user, identify which parts of the database may be of interest for the corresponding data analysis task, and recommend queries that retrieve relevant data. They treat SQL queries as items in the collaborative filtering approach, and recommend similar queries to related users.



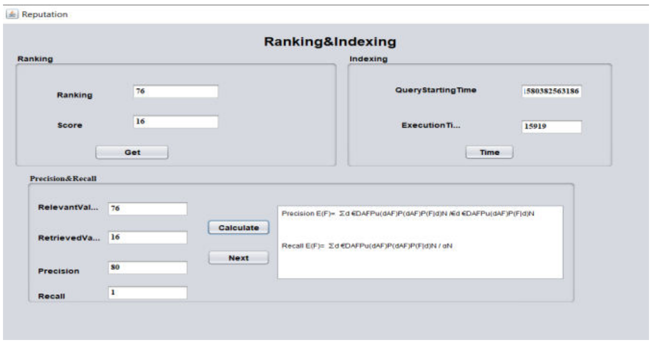
QueryAbout author

UserSearch

AuthorName ram

Fig(3): Query Recommendation

4.Indexing technique Retrieval: The indexing technique it is query optimization method and reduces the execution time also enhances the efficiency. Our experimental results show that our sampling approach never misses any true patterns when the number of feature instances.



Ranking&Indexing

Ranking

Ranking 76

Score 16

Indexing

QueryStartingTime 1580382563186

ExecutionTi... 15919

Precision&Recall

RelevantVal... 76

RetrievedVa... 16

Precision 80

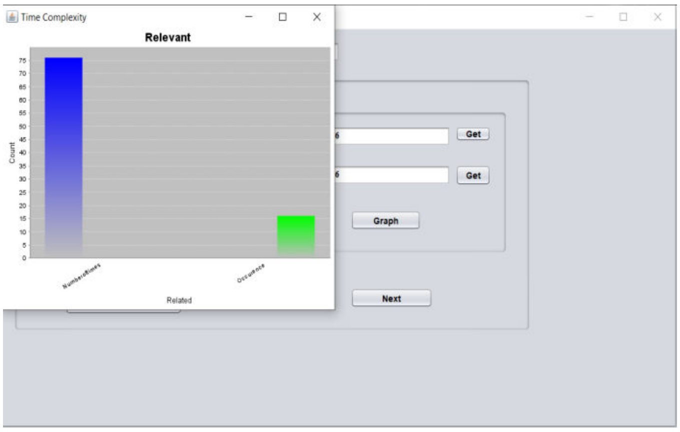
Recall 1

Precision E(F)= Σd €DAFPu(dAF)P(dAF)P(F|d)N /€d €DAFPu(dAF)P(F|d)N

Recall E(F)= Σd €DAFPu(dAF)P(dAF)P(F|d)N / σN

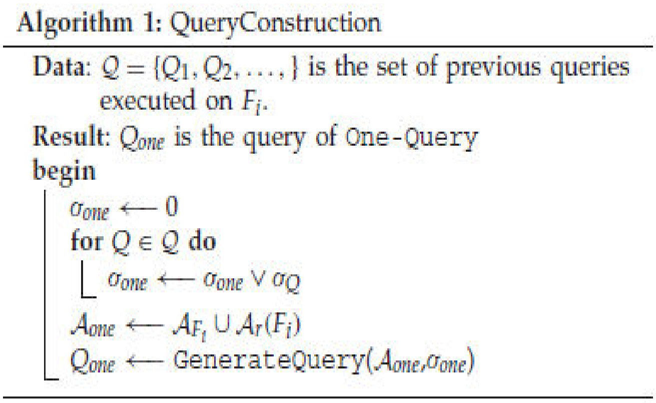
fig(4): Indexing and ranking page

5.Performance Analysis: Experimental results show that the dynamic approach often leads to higher success rate and simpler query forms compared with a static approach. The performance metrics which shows the graphical comparison of existing and proposed approach. Based on the query form the database engine which retrieves the result to the user. The retrieved result which is ranked based on the user query selection.



Fig(5): Performance analysis

4.METHODOLOGY:



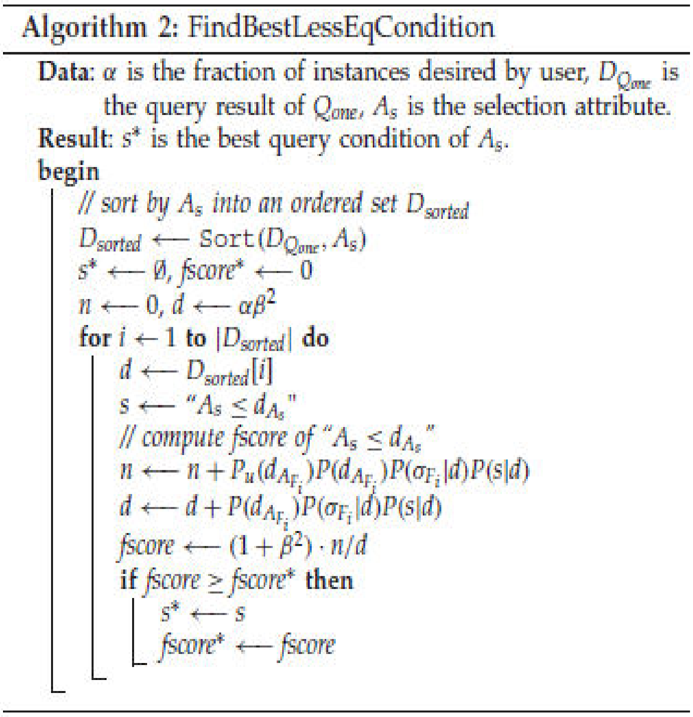
```
Algorithm 1: QueryConstruction
Data: $\mathcal{Q} = \{Q_1, Q_2, \ldots, \}$ is the set of previous queries
      executed on $F_i$.
Result: $Q_{one}$ is the query of One-Query
begin
    $\sigma_{one} \longleftarrow 0$
    for $Q \in \mathcal{Q}$ do
        $\sigma_{one} \longleftarrow \sigma_{one} \vee \sigma_Q$
    $\mathcal{A}_{one} \longleftarrow \mathcal{A}_{F_i} \cup \mathcal{A}_r(F_i)$
    $Q_{one} \longleftarrow$ GenerateQuery($\mathcal{A}_{one}, \sigma_{one}$)
```

Algorithm 1 describes the algorithm of the One-Query's query construction. The function Generate Query is to generate the database query based on the given set of projection attributes *Aone* with selection expression *σone*. When the system receives the result of the query *Qone* from the database engine, it calls the second algorithm of the F Score of each query condition by scanning one pass of data instances.the FScore of each query condition by scanning one pass of data instances. There are 2 steps to do this.

1) First, we sort the values of As in the order of a1 ≤ a2 ≤ .... ≤ am, where m is the number of As's values. Let Daj denote the set of data instances in which As's value is equal to aj.

2) Then, we go through every data instance in the order of As's value. Let query condition sj = "As ≤ aj" and its corresponding FScore be fscorej.

According to Eq. (3), fscorej can be computed as
fscorej = (1 + β2) · nj/dj,
nj = d∈DQonePu(dAFi)P(dAFi)P(σFi|d)P(si|d),
dj = d∈DQoneP(dAFi)P(σFi|d)P(si|d) + αβ2.
For j > 1, nj and dj can be calculated incrementally:
nj = nj−1+d∈DajPu(dAFi)P(dAFi)P(σFi|d)P(sj|d),
dj = dj−1 + d∈DajP(dAFi)P(σFid)P(sj|d).
Algorithm 2 shows the pseudocode for finding the best "≤" condition.Complexity: As for other query

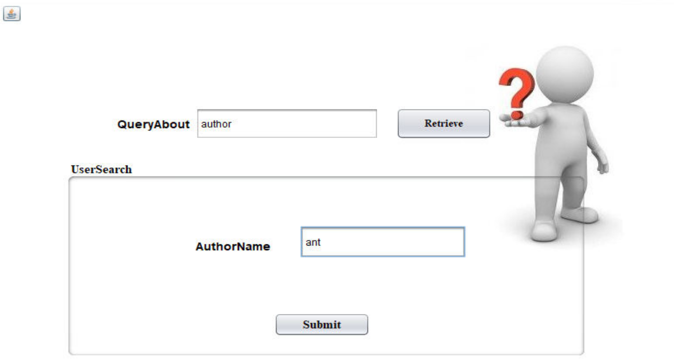


```
Algorithm 2: FindBestLessEqCondition
Data: $\alpha$ is the fraction of instances desired by user, $D_{Q_{one}}$ is
      the query result of $Q_{one}$, $A_s$ is the selection attribute.
Result: $s^*$ is the best query condition of $A_s$.
begin
    // sort by $A_s$ into an ordered set $D_{sorted}$
    $D_{sorted} \longleftarrow \mathtt{Sort}(D_{Q_{one}}, A_s)$
    $s^* \longleftarrow \emptyset$, $fscore^* \longleftarrow 0$
    $n \longleftarrow 0$, $d \longleftarrow \alpha\beta^2$
    for $i \leftarrow 1$ to $|D_{sorted}|$ do
        $d \longleftarrow D_{sorted}[i]$
        $s \longleftarrow$ "$A_s \leq d_{A_s}$"
        // compute fscore of "$A_s \leq d_{A_s}$"
        $n \longleftarrow n + P_u(d_{A_{F_i}})P(d_{A_{F_i}})P(\sigma_{F_i}|d)P(s|d)$
        $d \longleftarrow d + P(d_{A_{F_i}})P(\sigma_{F_i}|d)P(s|d)$
        $fscore \longleftarrow (1+\beta^2) \cdot n/d$
        if $fscore \geq fscore^*$ then
            $s^* \longleftarrow s$
            $fscore^* \longleftarrow fscore$
```

Algorithm 2 shows the pseudocode for finding the best "<=" conditionTwo selection components may have a lot of overlap (or redundancy). For example, if a user is interested in some customers with age between 30 and 45, then two selection components: "age > 28" and "age > 29" could get similar FScores and similar sets of data instances. Therefore, there is a redundancy of the two selections. Besides a high precision, we also require the recommended selection components should have a high diversity. diversity is a recent research topic in recommendation systems and web searchengines [6], [28]. However, simultaneously maximizing the precision and the diversity is an NP-Hard problem [6]. It cannot be efficiently implemented in an interactive system. In our dynamic query form system, we observe that most redundant selection components are constructed by the same attribute. Thus, we only recommend the best selection component for each attribute.

5.EXPERIMENTAL RESULT

Query recommendation page: This is an query recommendation page. In which we enter the query of author and the name of author. When we click on retrieve tab there is author database in query about and next In author name tab we enter the author name and click on submit button.



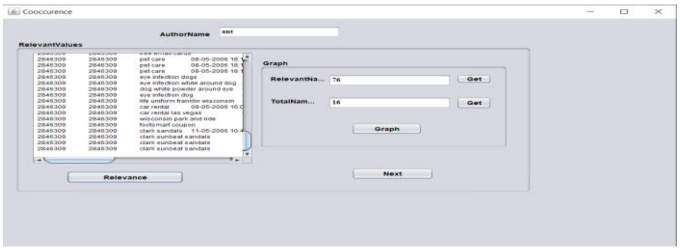
QueryAbout: author

UserSearch

AuthorName: ant

Fig(1):Query recommendation page

Relevance Page: In author name tab we enter the author name and click on relevance tab then all the data of that author name will be shown and also calculate total relevance and obtained relevance of that author name.



Cooccurence

AuthorName: ant

RelevantValues

```
2846309    2846309    pet care      08-05-2006 18:1
2846309    2846309    pet care      08-05-2006 18:1
2846309    2846309    pet care      08-05-2006 18:1
2846309    2846309    eye infection dogs
2846309    2846309    eye infection white around dog
2846309    2846309    dog white powder around eye
2846309    2846309    eye infection dog
2846309    2846309    life uniform franklin wisconsin
2846309    2846309    car rental    09-05-2006 16:0
2846309    2846309    car rental las vegas
2846309    2846309    wisconsin park and ride
2846309    2846309    footsmart coupon
2846309    2846309    clark sandals     11-05-2006 10:4
2846309    2846309    clark sunbeat sandals
2846309    2846309    clark sunbeat sandals
2846309    2846309    clark sunbeat sandals
```

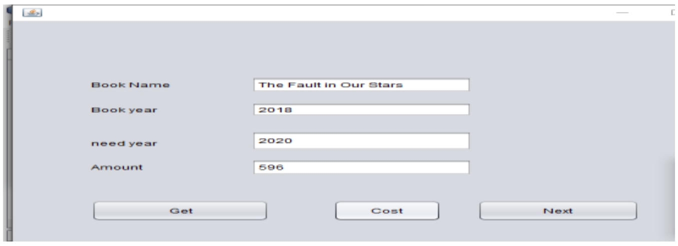
Graph

RelevantNa...: 76

TotalNam...: 16

Fig(2):Relevance Page

Prediction Page: Here we calculate prediction of books by using various parameters like year, cost and feedback. This process is iteratively run until the user satisfied with the result. By using user feedback system knows whether the user is satisfied or not. In these page we enter the book name, book year, need year and amount and click on get tab.



Book Name: The Fault in Our Stars

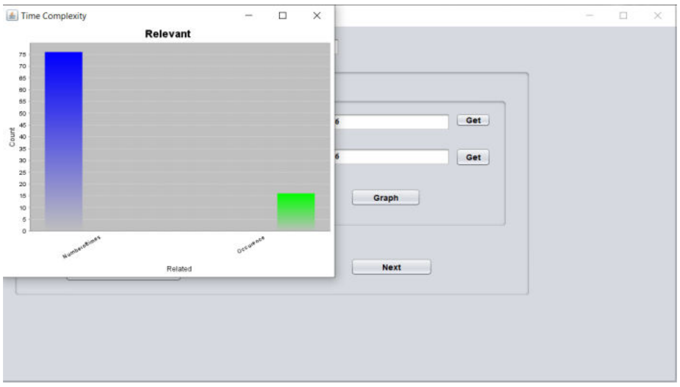
Book year: 2018

need year: 2020

Amount: 596

Fig(3):Prediction Page

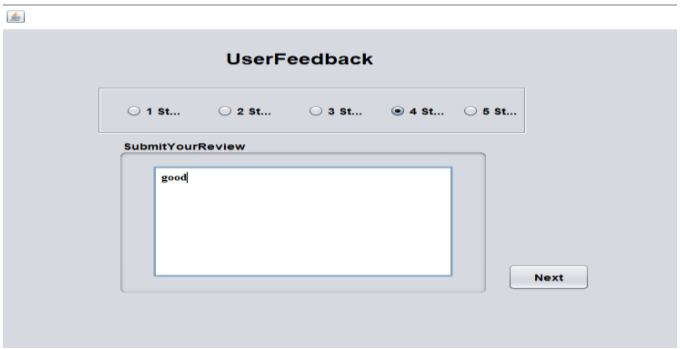
Relevance Graph: This is a graph of relevance and occurance.



Fig(4):Relevance Graph

User Feedback: This is a user feedback page. In that user can give the feedback after that system understand whether the user is satisfied with the result or not. If the user is not satisfied then the process will iteratively perform until he/she satisfied

.



good

Fig(5):User Feedback

6.CONCLUSION

In this paper we propose a dynamic query form generation approach which helps users dynamically generate query forms. The key idea is to use a probabilistic model to rank form components based on user preferences. We capture user preference using both historical queries and run-time feedback such as click-through. Experimental results show that the dynamic approach often leads to higher success rate and simpler query forms compared with a static approach. The ranking of form components also makes it easier for users to customize query forms.

7.REFERENCES

[1] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In Proceedings of VLDB pages 81–92, Berlin, Germany, September 2003.

[2] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diver-sifying search results. In Proceedings of WSDM , pages 5–14 Barcelona, Spain, February 2009.

[3] S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis. Automate ranking of database query results. In CIDR, 2003.

[4]G.Salton and M.McGill,"Introduction To Modern Informational Retrival",McGraw-Hill,1984

[5]L.Tang,T,Li,Y.Jiang,Z.Chen,"Dynamic Query Forms For Database Queries",IEEE transaction on knowledge and data engineering,2013

[6]G.Wolf,H.Khatri,B.Chokshi,J.Fan,Y.chan,and S.Kambhampati.Query proceeing over incomplete autonomous database.In Proceedingf of VLDB,pages 651-662,2007

[7]S.Cohen-Boulakia,O.Biton,S.Davidson,and C.Froidevaux,"Bioguidesrs:querying multiple sources with a user-centric perspective",Bioinformatics,2007.

[8]T.Zhang,R.Ramakrishnan,and M.Livny,"Birch: An efficient data clustering method for very large databases",in Proceedings of SIGMOD,1996.